

Entwurf einer Integrationsschnittstelle für Content-Management-Systeme basierend auf dem Datenmodell BMF

Diplomarbeit im Studiengang Audiovisuelle Medien
an der Hochschule der Medien, Stuttgart

vorgelegt von

Marius Hofmeister

am 28. September 2006

Erstprüfer : Prof. Dipl.-Ing. Uwe Schulz
Zweitprüfer : Dipl.-Ing. Hubert Hubbes (IRT)

Hochschule der Medien, Stuttgart
Fachbereich Electronic Media
D-70569 Stuttgart

Institut für Rundfunktechnik, München
Forschungs- und Entwicklungsinstitut der öffentlich-rechtlichen
Rundfunkanstalten in der BRD, Österreich und der Schweiz
D-80939 München

Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe und daß ich sämtliche Quellen im Text und im Literaturverzeichnis angegeben habe.

München, den 28. September 2006

.....

Marius Hofmeister

Kurzfassung

Thema der vorliegenden Diplomarbeit ist der Entwurf einer Integrationsschnittstelle für Content-Management-Systeme basierend auf dem Datenmodell BMF.

Konkret soll für das Content-Management-System (CMS) des Bayerischen Rundfunks (BR) eine Softwareschnittstelle entworfen werden, die auf dem Datenmodell Broadcast Metadata exchange Format (BMF) basiert. Die Schnittstelle soll Daten aus dem CMS ausgeben und in die BMF-Struktur transformieren.

Das CMS des BR beinhaltet u.a. Metadaten der Fernsehproduktionsumgebung (z.B. den Namen eines beteiligten Kameramanns oder den technische Aufbau einer Sendung) in einer nicht veröffentlichten Datenstruktur. Bekannt ist, daß diese Struktur auf FESAD, der Fernseharchivdatenbank der öffentlich-rechtlichen Rundfunkanstalten, basiert, und daß CMS-Daten über das proprietäre AXF-Format ausgetauscht werden können.

Um einen standardisierten Metadatenaustausch zwischen Rundfunkanstalten und Applikationen (z.B. Schnittsystemen) zu ermöglichen, wurde das Datenmodell BMF entwickelt, das die Struktur von Metadaten der Fernsehproduktion festlegt. Für BMF ist diese Diplomarbeit ein erster Praxistest.

Zunächst wurden objektive Anforderungen an die Schnittstelle gesammelt und der tatsächliche Informationsgehalt von FESAD mit dem des CMS verglichen, da dessen Datenstruktur wie erwähnt nicht veröffentlicht ist. Insgesamt zeigte sich eine große Übereinstimmung beider Modelle, es gab jedoch auch Unterschiede. Diese Arbeit behandelt daher nicht nur die Abbildung des unternehmensinternen BR-Datenmodells auf BMF, sondern auch die des ARD-weiten Datenmodells FESAD.

Nach Abgrenzung des abzubildenden Informationsgehaltes wurde geprüft, inwiefern sich dieser im derzeit aktuellen BMF-Klassenmodell wiederfindet. Im Zuge dessen wurden einige BMF-Pakete neu modelliert.

Der letztendliche Entwurf der Schnittstelle umfasst die Spezifikation von Zugriffsmethoden und die Datentransformation von CMS- bzw. FESAD-Daten auf BMF.

Als Ergebnis dieser Arbeit kann festgehalten werden, daß die vom BR gestellten Anforderungen an die Schnittstelle erfüllt und das BR-Datenmodell bzw. FESAD auf BMF abgebildet werden konnten. Dieser Diplomarbeit kann nun eine Implementierung der Schnittstelle folgen.

Vorwort

Die vorliegende Diplomarbeit entstand am Institut für Rundfunktechnik (IRT), dem Forschungs- und Entwicklungsinstitut der öffentlich-rechtlichen Rundfunkanstalten in der Bundesrepublik Deutschland (ARD/ZDF/DLR), Österreich (ORF) und der Schweiz (SRG/SSR). Der Entwurf und die Implementierung der vorliegenden Schnittstelle erfolgt in einem Projekt, welches das IRT in Zusammenarbeit mit dem Bayerischen Rundfunk (BR) und dem Medienberatungsunternehmen Flying Eye durchführt.

Danksagungen

Da diese Diplomarbeit Teil des beschriebenen Projektes ist, war ihr Gelingen zum großem Teil von einer gut funktionierenden Kooperation aller Beteiligten abhängig. Häufig galt es, Informationen zu den verwendeten Datenmodellen zu recherchieren. In diesem Zusammenhang gilt mein Dank Wolfgang Englmaier, dem Leiter der „Systemplanung Archive“ der Abteilung „Technische Planung und Infrastruktur“ des BR München, der stets für alle aufkommenden Fragen zur Verfügung stand und eine überaus unkomplizierte Zusammenarbeit ermöglichte.

Danken möchte ich auch den Kollegen von Flying Eye, besonders Arne Distelrath, für die sehr gute Kooperation.

Meinem Betreuer am IRT, Hubert Hubbes, gilt ganz besonderer Dank, da er mir während der gesamten Bearbeitungszeit stets unterstützend und aufbauend zur Seite stand.

Des weiteren bedanke ich mich bei meinem betreuenden Hochschulprofessor Uwe Schulz, dem BMF-Projektleiter Andreas Ebner und meinen sorgfältigen Korrekturlesern.

Aufbau der Arbeit

- | | |
|------------------|---|
| Kapitel 1 | gibt eine Einleitung in den Themenkomplex. |
| Kapitel 2 | befasst sich mit den für das Verständnis der Arbeit notwendigen theoretischen Grundlagen. |
| Kapitel 3 | gibt ein Überblick über das digitale Archiv des BR und die damit verbundenen Datenmodelle FESAD und AXF. |
| Kapitel 4 | gibt eine Einführung in das vom IRT entwickelte Datenmodell BMF und die für diese Arbeit relevanten Pakete. |
| Kapitel 5 | listet die Anforderungen auf, die an die zu entwerfende Schnittstelle gestellt werden. |
| Kapitel 6 | spezifiziert die zu entwickelnde Schnittstelle, indem Zugriffsmethoden und Datentransformation besprochen werden. |
| Kapitel 7 | zieht Bilanz der Diplomarbeit und wagt einen Ausblick in die Zukunft. |

Typographische Konventionen

Die aus den Datenmodellen stammenden Fachbegriffe dieser Arbeit werden kursiv dargestellt; alle anderen Fachbegriffe nicht kursiv.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 2 | Grundlagen | 3 |
| 2.1 | Metadaten in der IT-basierten Fernsehproduktion | 3 |
| 2.2 | Datenmodelle | 5 |
| 2.3 | Referenzierungen | 5 |
| 3 | Das digitale Archiv des Bayerischen Rundfunks | 8 |
| 3.1 | Aufbau | 8 |
| 3.2 | Verwendetes Metadatenmodell | 10 |
| 3.2.1 | Fernseharchivdatenbank (FESAD) | 11 |
| 3.2.1.1 | Aufbau | 11 |
| 3.2.1.2 | Hierarchieebenen | 12 |
| 3.2.2 | Attribute eXchange Format (AXF) | 13 |
| 3.2.3 | Abweichungen zu FESAD im BR-Modell | 14 |
| 3.2.3.1 | Hierarchieebenen | 14 |
| 3.2.3.2 | Straten | 15 |
| 3.2.3.3 | Order | 16 |
| 3.3 | Schnittstellenarchitektur | 17 |
| 4 | Broadcast Metadata exchange Format (BMF) | 18 |
| 4.1 | Entwicklung | 18 |
| 4.2 | Aufbau | 20 |
| 4.2.1 | Redaktioneller und technischer Teil | 20 |

| | | |
|----------|--|-----------|
| 4.2.2 | Hierarchieebenen | 21 |
| 4.3 | Klassenmodell | 22 |
| 4.3.1 | Paket Programme | 23 |
| 4.3.1.1 | Productionelements | 23 |
| 4.3.1.2 | Grouping | 25 |
| 4.3.1.3 | Participant | 26 |
| 4.3.1.4 | Right | 27 |
| 4.3.1.5 | Programmplanung | 27 |
| 4.3.2 | Paket Production | 28 |
| 4.3.2.1 | ContentInstance | 28 |
| 4.3.2.2 | Planning | 30 |
| 4.3.2.3 | PEState | 31 |
| 4.3.3 | Paket Publishing | 32 |
| 4.3.3.1 | Transmission | 32 |
| 4.3.4 | Paket Annotation | 33 |
| 4.3.4.1 | Sequence | 33 |
| 4.3.4.2 | AdditionalInformation | 34 |
| 4.3.4.3 | Award | 36 |
| 4.3.4.4 | Category | 36 |
| 4.3.4.5 | MainDescriptor | 37 |
| 5 | Anforderungen an die Schnittstelle | 38 |
| 6 | Entwurf der Schnittstelle | 40 |
| 6.1 | Zugriffsmethoden | 40 |
| 6.1.1 | getPEFull | 41 |
| 6.1.2 | getPEStandard | 41 |
| 6.1.3 | getRightsequence | 41 |
| 6.1.4 | getFullOrder | 42 |
| 6.1.5 | getStandardOrder | 42 |
| 6.2 | Datentransformation | 43 |
| 6.2.1 | Abbildung DU auf SinglePEFull / SinglePEStandard | 46 |

| | | |
|----------|---|-----------|
| 6.2.1.1 | Hierarchische Abbildung (CONTAINER, Link) | 47 |
| 6.2.1.2 | Dokumentarische Einheit (DU, Automationsdatensatz) . . | 49 |
| 6.2.1.3 | Titel (TITLE) | 54 |
| 6.2.1.4 | Urheber, Produzent, Mitwirkende (APC) | 56 |
| 6.2.1.5 | Inhaltsbeschreibung (CONTENT) | 59 |
| 6.2.1.6 | Kategorie (CATEGORY) | 62 |
| 6.2.1.7 | Hauptdeskriptor (MD) | 63 |
| 6.2.1.8 | Anmerkung (REMARK) | 64 |
| 6.2.1.9 | Lokale Zusatz-Info (INFO) | 66 |
| 6.2.1.10 | Verknüpfung (LINK) | 68 |
| 6.2.1.11 | Rechte-Informationen (LICENCE) | 69 |
| 6.2.1.12 | Beschränkung (RESTRICTION) | 70 |
| 6.2.1.13 | Identifikationsnummer (ID-OF-DU) | 71 |
| 6.2.1.14 | Ausstrahlungs- und Planungsdaten (BC) | 72 |
| 6.2.1.15 | Urheber, Produzent, Mitwirkende der Wiederholung (RAPC) | 76 |
| 6.2.2 | Abbildung Rechtesequenz auf Sequence | 77 |
| 6.2.2.1 | Beschränkung (RESTRICTION) | 77 |
| 6.2.2.2 | Zeitliche Einbindung | 79 |
| 6.2.2.3 | Gültigkeitsbereich (Bild oder Ton) | 80 |
| 6.2.2.4 | Zuordnung der Sequenz zu übergeordnetem Element . . . | 80 |
| 6.2.2.5 | Referenzierung auf Lizenzobjekte | 80 |
| 6.2.3 | Abbildung Order auf FullOrder / StandardOrder | 82 |
| 7 | Zusammenfassung und Ausblick | 88 |
| | Literaturverzeichnis | 89 |
| | Abbildungsverzeichnis | 90 |
| | Tabellenverzeichnis | 92 |

Kapitel 1

Einleitung

Arbeitsabläufe der Fernsehproduktionstechnik zeichnen sich häufig durch eine hohe Komplexität und über viele Jahre gewachsene Strukturen aus. Oft ist eine Vielzahl von Systemen unterschiedlicher Hersteller im Einsatz, deren Zusammenwirken immer neue Herausforderungen an die Planer stellt.

Durch die ständig wachsende Leistungsfähigkeit der Informationstechnik (IT) findet jedoch zur Zeit ein Umbruch in der Gestaltung der Arbeitsabläufe von Rundfunkanstalten statt. Immer mehr Prozesse und Anwendungen können kostengünstiger und effektiver durch IT-Systeme realisiert werden. Schon heute produzieren führende Fernseh- und Rundfunkanstalten rein digital.

Der Austausch von Daten nimmt darin die zentrale Rolle ein. Doch es geht bei Weitem nicht nur um die Weitergabe von reinen Nutzdaten, Essenz genannt, die beispielsweise ein Video darstellen, sondern auch um eine nähere Beschreibung dieser Essenz (z.B. Name des beteiligten Kameramanns, technischer Aufbau einer Sendung oder Rechteverwertungsinformationen). Solche Daten, die andere Daten beschreiben, nennt man Metadaten.

Bislang liegen Metadaten in den Rundfunkanstalten in unterschiedlicher Struktur vor, so daß ein Austausch nicht ohne Weiteres möglich ist. Aus diesem Grund entwickelte das IRT das Broadcast Metadata exchange Format (BMF), welches diesen Austausch spezifiziert. In vorliegender Diplomarbeit soll nun ein erster Schritt zur praktischen Anwendung von BMF getan werden.

Für das Content-Management-System (CMS) des Bayerischen Rundfunks (BR), das u.a. Metadaten der Fernsehproduktionsumgebung beinhaltet, soll eine Integrationsschnittstelle zu BMF entwickelt werden, die nach dieser Arbeit implementiert, also umgesetzt, werden soll. Mittels dieser Schnittstelle sollen Metadaten des CMS angefordert und konform zum Datenmodell BMF ausgegeben werden können, um dann von externen Anwendungen (z.B. Schnittsystemen) genutzt und weiterverarbeitet werden zu können. Voraussetzung dafür ist natürlich, daß diese Anwendungen ebenfalls BMF als Schnittstelle implementieren.

Es geht in dieser Arbeit jedoch nicht nur um eine den spezifischen Anforderungen des BR angepaßte Schnittstelle, da das Metadatenmodell des BR auf der Fernseharchivdatenbank FESAD basiert, die ARD-weit im Einsatz ist. Sie beinhaltet Metadaten einer großen Anzahl öffentlich-rechtlicher Produktionen und dient Redakteuren damit als Recherchemöglichkeit für Bild- und Tonmaterial von Partneranstalten.

Daher behandelt diese Diplomarbeit zwar die Erstellung einer Schnittstelle für das CMS des BR; weite Teile der Ausführung gelten jedoch ebenso für das weit verbreitete FESAD-Datenmodell.

Kapitel 2

Grundlagen

Das vorliegende Kapitel enthält grundlegende Informationen, die für das weitere Verständnis der Arbeit von Bedeutung sind. Angesprochen wird die prinzipielle Kategorisierung von Metadaten in Fernsehproduktionsumgebungen, die Verwendung von unterschiedlichen Datenmodellen in dieser Arbeit sowie spezielle Mechanismen der objektorientierten Programmierung, deren Grundlagenkenntnis jedoch vorausgesetzt wird.

2.1 Metadaten in der IT-basierten Fernsehproduktion

Die Anforderungen, denen Metadaten im Rundfunk genügen müssen, definieren sich durch die bekannten Arbeitsprozesse im Fernsehen: es gilt, Produktion, Verteilung, Sendung und Archivierung des Inhalts innerhalb einer Produktionsumgebung zu beschreiben. Mit jedem Produktionsschritt entstehen dabei neue Informationen, die auf die Essenz verweisen.

Abb. 2.1 zeigt einen beispielhaften Fernsehproduktionsworkflow und der zugehörige Umgang mit Metadaten. Wo früher begleitende papierbasierte Dokumente, z.B. MAZ-Karten¹, protokolliert und mit den Medien mitgeführt wurden, können diese Informationen heute digital in zentralen Datenbanken oder zusammen mit der Essenz abgespeichert werden und stehen so fortwährend zur Verfügung.

Im TV-Bereich lassen sich Metadaten daher definieren als alle „erforderlichen Daten“,

¹MAZ: **M**agnetische **B**ild **A**ufzeichnung

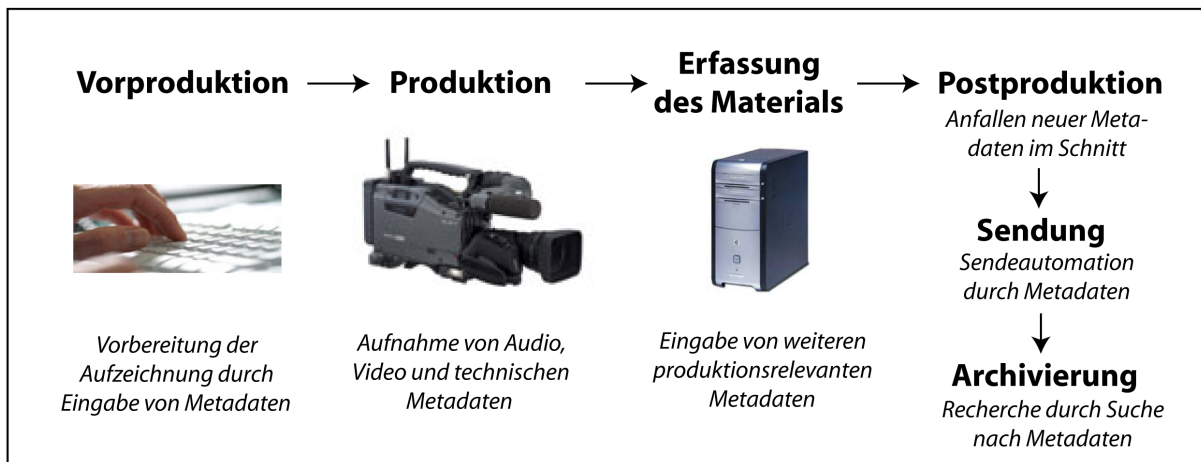


Abbildung 2.1: Metadaten in einem beispielhaften Fernsehproduktionsworkflow

die eine effiziente und weitgehend automatisierte Verarbeitung von Nutzdaten (Video, Audio, Zusatzdaten), deren Verwaltung und Speicherung sowie den Austausch zwischen beteiligten Verarbeitungsprozessen und den Nutzern ermöglichen. [4]

Anwendung finden Metadaten hauptsächlich in der Lokalisierung, Verwendung und Verwaltung von Essenz, sowie im Verwertungs- und Rechte-Management. [1]

Metadaten lassen sich grob in zwei Gruppen einteilen (siehe [9]):

- **Technische Metadaten** beschreiben die technische Struktur von Essenz, d.h. Informationen zu deren Zusammensetzung, Signal- und Speichereigenschaften. Beispielsweise sei hier die Bit-Tiefe eines Videosignals genannt oder die Anzahl der Audiokanäle in einem Videofile.
- **Deskriptive Metadaten** beschreiben die redaktionell-inhaltliche Struktur von Essenz, d.h. beispielsweise Angaben zu den Teilnehmern einer Produktion oder Verschlagwortung für eine spätere Volltextsuche unter Essenzelementen.

2.2 Datenmodelle

Im Verlauf dieser Arbeit ist die Rede von Datenmodellen, zwischen denen ein Mapping² stattfindet. Der Begriff Datenmodell repräsentiert dabei eine Menge von Entitäten³ und deren Beziehungen untereinander.

Im vorliegenden Fall geht es um unterschiedliche Formen von Datenmodellen. Das an FE-SAD orientierte Datenmodell des BR liegt als Entity Relationship Diagram (ERD) vor, welches vor allem im Bereich von relationalen Datenbanken Anwendung findet. Es liefert Informationen über die eingebundenen Elemente, Tabellen genannt, und deren Beziehungen untereinander.

Das Datenmodell BMF lag zu Beginn als ERD-Entwurf vor und wird gerade Stück für Stück in ein objektorientiertes Modell überführt. Objektorientierte Modelle geben zusätzlich zu den in einem ERD vorhandenen Informationen Auskunft über Vererbungsbeziehungen sowie Aggregations- und Kompositionsbeziehungen (siehe Kap. 2.3) zwischen Objekten.

ERD-Tabellen finden ihre Entsprechung in den Klassen des objektorientierten Modells. Einzelne Datensätze der ERD-Tabellen („Stammdaten“) finden ihre Entsprechung in Objektinstanzen einzelner Klassen.

2.3 Referenzierungen

Zum Verständnis der folgenden Klassenmodelle sei hier der Unterschied zwischen den Referenzierungsarten **Aggregation** (Weak-Reference) und **Komposition** (Strong-Reference) erläutert. Sie beschreiben die Beziehung von Klassen und Objekten untereinander. Eine Aggregation zeichnet aus, daß das abhängige Element (in Abb. 2.2 *Klasse1*) auch ohne das Gegenüber existieren kann. In einer Komposition hingegen setzt eine Existenz des abhängigen Elementes immer die Existenz des Gegenübers voraus.

Das Element, welches an eine Raute grenzt (in Abb. 2.2 *Klasse2*), ist stets das referen-

²Mapping: Abbildung der Information eines Datenmodells auf ein anderes

³Entität: eindeutig zu bestimmendes Objekt, dem Informationen zugeordnet werden können

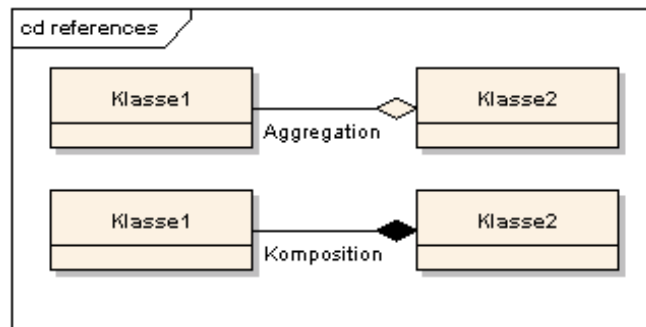


Abbildung 2.2: Aggregation und Komposition

zierende Element. Die farbliche Darstellung der Raute gibt Auskunft über die Referenzierungsart. Diese grafische Notation stützt sich auf den UML⁴-Standard.

Abb. 2.3 stellt eine umgekehrte Referenzierung dar, die in BMF häufig angewendet wird. Erkennbar sind eine Klasse *Programme* (Sendung) und ihre zugehörige Klasse *ContentInstanceList*, die die technischen Metadaten einer Sendung enthält.

Im ersten Beispiel der Abbildung besteht zwischen den Klassen eine Komposition. Wer die Klasse *Programme* implementiert, muß in diesem Fall auch immer die Klasse *ContentInstanceList* implementieren. Ist dies nicht erwünscht, wie z.B. bei komplexeren Klassen, die nicht immer für die Verwendung der Klasse *Programme* relevant sind, ist es möglich, eine **umgekehrte Referenzierung** zu wählen. In diesem Fall referenziert die Klasse *ContentInstanceList* auf die Klasse *Programme* (Aggregation).

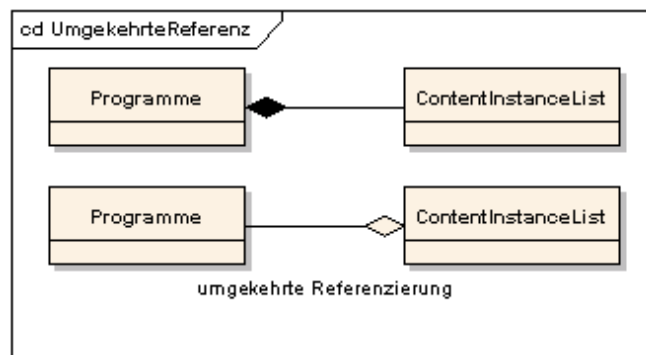


Abbildung 2.3: Referenzierungsarten

⁴UML: Unified Modeling Language, eine von der Object Management Group (OMG) entwickelte und standardisierte Sprache zur Modellierung von Software und anderen Systemen

Zur Folge hat dieser Mechanismus, daß bei Übertragung eines Objektes vom Typ *Programme* im Falle der umgekehrten Referenzierung etwaige Objekte vom Typ *ContentInstanceList* nicht mitgeliefert werden. Für die Praxis heißt das, daß für diesen Anwendungsfall eine Art Containerklasse implementiert werden muß, die beide Objekte enthält. In dieser Diplomarbeit kommt der Mechanismus vor allem in den Klassen *SinglePEFull* und *SinglePEStandard* (siehe Kap. 6.2.1) zum Vorschein.

Kapitel 3

Das digitale Archiv des Bayerischen Rundfunks

Das vorliegende Kapitel behandelt die Komponenten des digitalen BR-Archivs sowie das Metadatenmodell des BR-Content-Management-Systems. Im Zuge dessen wird auf das generische Austauschformat AXF eingegangen und die Fernseharchivdatenbank FESAD. Abweichungen des BR-Datenmodells zu FESAD werden analysiert und dargestellt.

Das Kapitel schließt mit der Schnittstellenarchitektur des CMS und stellt die Anbindung der in dieser Arbeit entworfenen Schnittstelle dar.

3.1 Aufbau

Zu den Komponenten des BR-Archivs gehören ein Zentral- und Archivspeicher, ein Content-Management-System¹ (CMS), ein Automationssystem und eine FESAD-Datenbank (siehe Abb. 3.1).

Der Zentralspeicher fasst bis zu 1600h IMX²-Material. Er stellt Video- und Audiodaten zum Playout³ bereit und fungiert auch als Zwischenspeicher für das Archivsystem, welches mittels zweier Bandroboter das bereits digitalisierte Material auf Magnetbändern sichert.

¹Content-Management-System (CMS): Computergestütztes System zur Verwaltung von Daten

²IMX: Interoperability Material Exchange, Videoformat der Firma Sony, Datenrate 50MBit/s

³Playout: Ausstrahlung von Audio- und Videomaterial

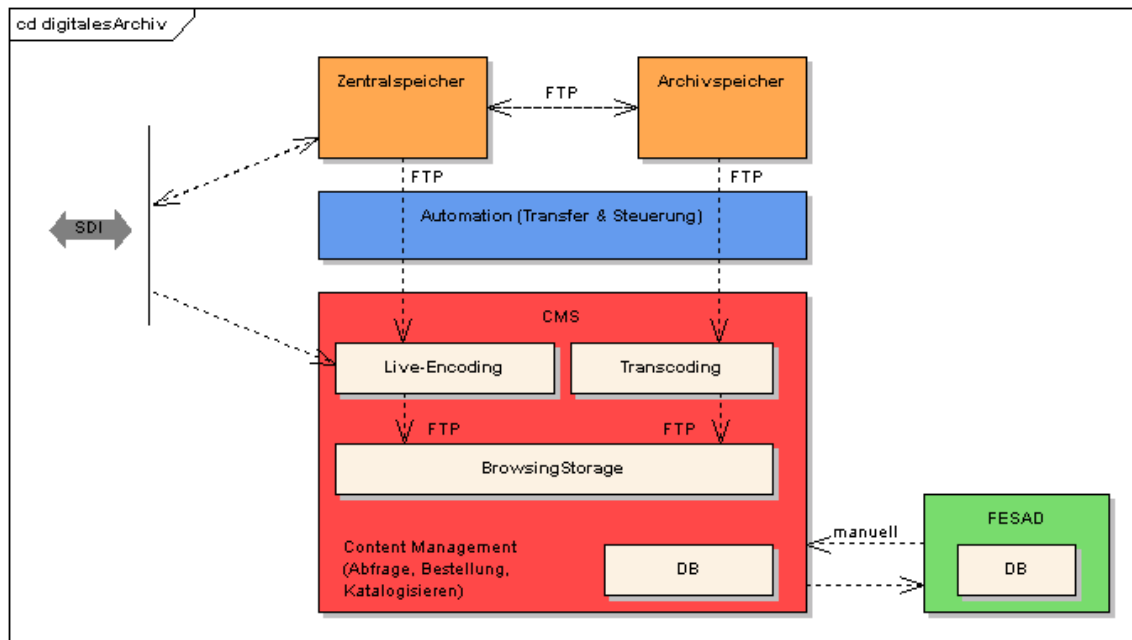


Abbildung 3.1: Das digitale Archiv im BR (siehe [6])

Ziel ist es, sukzessive das gesamte Programmvermögen des BR zu digitalisieren und im Bandarchiv zu sichern.

Das CMS der Firma Blue Order umfasst eine Datenbank für technische und deskriptive Metadaten sowie ein Speichersystem für „Low-Resolution“-Kopien der Essenz (sog. Browsingkopien), die Redakteuren als Sichtungskopien dienen.

Der Ingest⁵ von Videomaterial erfolgt direkt auf den Zentralspeicher und parallel in „Low-Resolution“-Qualität in das CMS, wo ein Metadatensatz angelegt und mit Automationsdaten technischer Natur gefüllt wird.

Der Dokumentar hat die Möglichkeit, über Eingabemasken („Cataloging-Client“) Informationen in diesen Metadatensatz einzupflegen. Nach dem Ingest oder Playout kann er das Material aus dem Zentralspeicher per Mausklick in den Archivspeicher überführen.

Die Redakteure können dann nach Material recherchieren und es auswählen. Dabei ist das Sichten der „Low-Resolution“-Kopien und die Verschlagwortung von Videodaten gleichermaßen hilfreich. Nach einem Bestellvorgang und Freigabe durch das Archiv wird das so selektierte Material auf das jeweilige Produktionssystem (Videoserver) transferiert. Ne-

⁴Low-Resolution: Videodateien mit niedriger Datenrate, hier: 1,5MBit/s, MPEG1-komprimiert

⁵Ingest: Einspielung von Videomaterial auf Server-basierte Produktionsumgebungen

ben der reinen Essenz können auch im CMS erstellte Rohschnittlisten u.a. in Form einer AAF⁶-Datei bereitgestellt werden, die dann zur Weiterverarbeitung in Schnittsystemen zur Verfügung steht (siehe auch Kap. 3.2.3.3).

Zum Materialaustausch zwischen ARD-Anstalten wird eine FESAD-Datenbank gepflegt. Über einen Mausklick können Video-Metadaten aus dem CMS nach FESAD exportiert und anderen Rundfunkanstalten zur Verfügung gestellt werden. Redakteure kooperieren der Anstalten recherchieren in dieser Datenbank nach Videomaterial, das zur Weiterverwendung angefordert werden kann. Über einen manuellen Ex- und Import können FESAD-Datensätze auch in das CMS importiert werden.

3.2 Verwendetes Metadatenmodell

Das Metadatenmodell des CMS im BR setzt auf einer relationalen Datenbank auf und wurde aus firmeninternen Erwägungen der Herstellerfirma Blue Order nicht veröffentlicht, d.h. lag auch dem Autor dieser Diplomarbeit nicht vor. Es basiert jedoch auf FESAD, dem Fernseharchivdatenmodell, das in den öffentlich-rechtlichen Rundfunkanstalten entwickelt und über viele Jahre den Anforderungen dieser angepasst wurde.

Der Ex- und Import von CMS-Metadaten geschieht über das proprietäre AXF-Format, welches auf XML basiert (siehe Kap. 3.2.2). Um die tatsächliche Übereinstimmung des BR-Datenmodells zu FESAD zu überprüfen, wurden eine AXF-Beispieldatei und eine XSLT⁷-Datei, die die Transformation eines FESAD-Datenexportes nach AXF und damit ins BR-Datenmodell beschreibt, analysiert.

Das FESAD-Modell liegt als ERD vor, das AXF-Format als DTD⁸ (siehe [7]) und Beispieldatei. In den grafischen Abbildungen dieser Diplomarbeit wurden die Modelle jedoch als Objektdiagramme dargestellt, um einen einfachen Vergleich zu BMF zu ermöglichen. AXF repräsentiert dabei immer das BR-Modell mit dessen Eigenheiten.

⁶AAF: **A**dvanced **A**uthoring **F**ormat, Dateiformat zum plattformübergreifenden Austausch von Multimedia-Essenz und Metadaten, wird vorwiegend im Postproduktionsbereich genutzt

⁷XSLT: **e**Xtensible **S**tyle **L**anguage **T**ransformations, beschreibt und steuert die Transformation von XML-Dokumenten in andere Formate

⁸DTD: **D**ocument **T**ype **D**efinition, definiert die Struktur von XML-Dateien

3.2.1 Fernseharchivdatenbank (FESAD)

Es folgt eine Einführung in die Fernseharchivdatenbank FESAD, an der sich das Datenmodell des BR wesentlich orientiert. Einleitend sei angemerkt, daß es sich in dieser Diplomarbeit immer um die aktuelle Variante namens FESADneu handelt, wenn von FESAD die Rede ist. Detaillierte Informationen zum ERD-Modell finden sich in [8].

3.2.1.1 Aufbau

Im Mittelpunkt von FESAD steht eine Dokumentarische Einheit (DU^9), die eine Sendung, einen Beitrag oder eine Sequenz repräsentieren kann. Sie liegt als Datensatz in der Tabelle *DU* vor und „besitzt“ weitere auf sie referenzierende Datensätze anderer Tabellen, die deskriptive und technische Metadaten enthalten. Je Tabelle können so (0..n) auf eine *DU* bezogene Datensätze existieren.

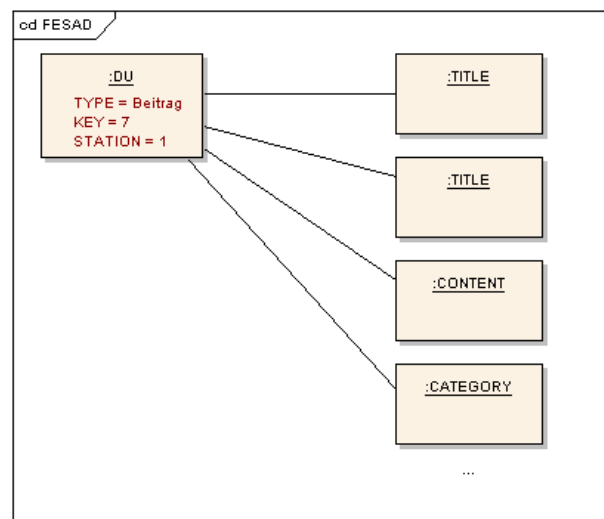


Abbildung 3.2: Beispielhafter Auszug der FESAD-Datenstruktur

Für diese Diplomarbeit wurden 15 FESAD-Tabellen berücksichtigt. Die Tabellen der technischen Metadaten *LOCATOR*, *SOUNDTRACK*, *MEDIUM*, *CONDITION*, *ID_OF_MEDIUM* und *STATION* wurden vom BR aus dieser Diplomarbeit ausgenommen.

⁹DU: **D**ocumentary **U**nit (engl.)

Abb. 3.2 stellt den Aufbau des FESAD-Datenmodells beispielhaft dar. Ein Datensatz vom Typ *DU* „besitzt“ Datensätze anderer Tabellen über (0..n)-Beziehungen. Dies ist angedeutet durch Verweise auf Datensätze der Tabellen *TITLE*, *CONTENT* und *CATEGORY*.

3.2.1.2 Hierarchieebenen

In FESAD ist es möglich, hierarchische Zusammensetzungen unter Dokumentarischen Einheiten darzustellen. So kann eine *DU* vom Typ Sendung aus (0..n) Beiträgen bestehen und ein Beitrag wiederum aus (0..n) Sequenzen. Die Einbindung einer *DU* in eine Übergeordnete geschieht mittels der Tabelle *CONTAINER*. Dort werden Position (*TC*) und Dauer (*DURATION*) der eingebundenen *DU* sowie die *DU*-Elemente selbst (*PARENT_KEY*, *CHILD_KEY*) angegeben.

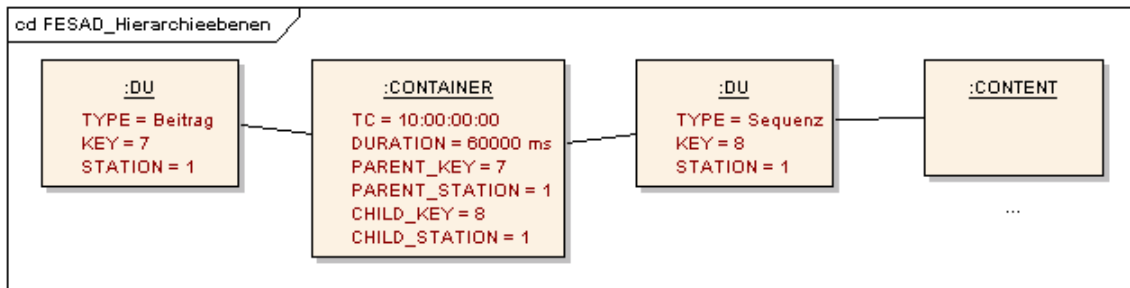


Abbildung 3.3: Hierarchieebenen in FESAD

In Abb. 3.3 besitzt eine *DU* vom Typ Beitrag eine *DU* vom Typ Sequenz, die über einen festgelegten Zeitabschnitt deskriptive Metadaten (z.B. Angaben über im Bild erscheinende Drehorte) in der Tabelle *CONTENT* einbindet. Die Dauer des Zeitabschnitts, für den die Metadaten gelten, wird in der Tabelle *CONTAINER* angegeben. Dort werden auch zwei zusammengehörige Dokumentarische Einheiten verknüpft. Durch Kombination eines für die Rundfunkanstalt spezifischen Schlüssels (*STATION*) und eines *DU*-spezifischen Schlüssels (*KEY*) läßt sich eine *DU* eindeutig bestimmen.

3.2.2 Attribute eXchange Format (AXF)

Es folgt eine kurze Einführung in das AXF-Format der Firma Blue Order, sofern es für das Verständnis dieser Diplomarbeit von Bedeutung ist. Bei AXF handelt sich um ein generisches Austauschformat auf XML-Basis. Detaillierte Informationen über AXF finden sich in [7].

Abb. 3.4 zeigt beispielhaft ein Objekt vom Typ *DU* als AXF-Instanz. *AXFRoot* ist der Hauptknoten, der beliebig viele Kindelemente vom Typ *MAObject* besitzen kann, jedoch nur ein *MAObject* vom Typ *DU* (*mdclass*=“*DU*“). Alle anderen Elemente vom Typ *MAObject* sind entweder sich auf die *DU* beziehende Unterelemente (z.B. *mdclass*=“*TITLE*“) oder besitzen das Attribut *VDX* (*mdclass*=“*VDX*“) und stellen die Straten der *DU* dar (zu Straten siehe Kap. 3.2.3).

Das Element *MAObject* kann Kindelemente vom Typ *Ref* und *Meta* besitzen. Während in *Meta* die zu übermittelnde Information direkt vorliegt, werden in *Ref* Beziehungen zwischen Elementen vom Typ *MAObject* ausgewertet.

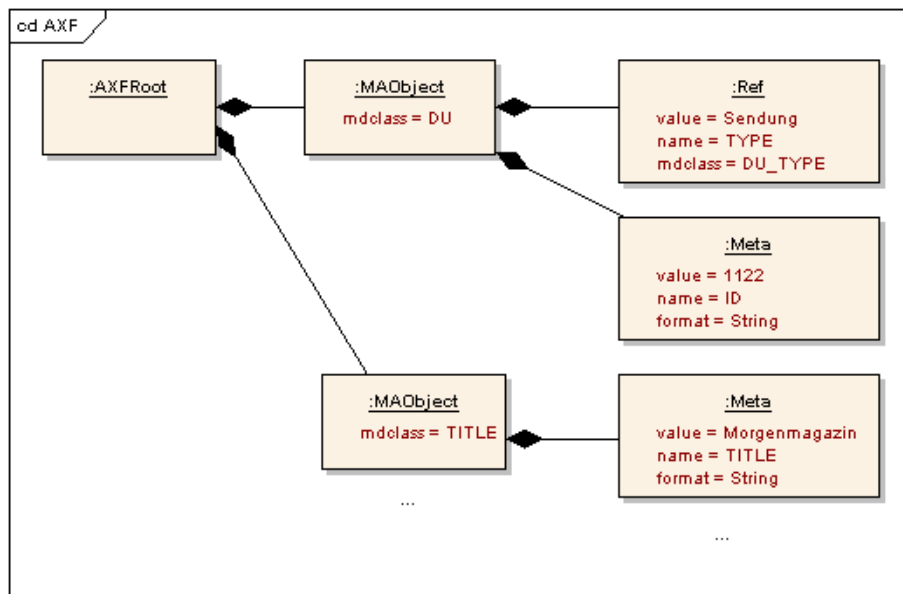


Abbildung 3.4: Beispielhafter Auszug der AXF-Datenstruktur

Beispielhaft sei hier der zu Abb. 3.4 korrespondierende Quelltext in XML-Kodierung dargestellt. Aus Übersichtlichkeitsgründen wird im Laufe dieser Diplomarbeit auf die XML-Darstellung verzichtet.

```
<AXFRoot>
  <MAObject mdclass="DU">
    <Ref name="TYPE" mdclass="DU_TYPE"> 1 </Ref>
    <Meta format="String" name="ID"> 1122 </Meta>
  </MAObject>
  <MAObject mdclass="TITLE">
    <Meta format="String" name="TITLE"> Morgenmagazin </Meta>
  </MAObject>
</AXFRoot>
```

Im XML-Knoten *Ref* wird auf ein *Meta*-Element eines *MAObject* (*mdclass*="DU_TYPE") mit dem Codewert 1 verwiesen. Durch diese Objektbeziehung kann der eigentliche Wert von *TYPE* ermittelt werden, im Beispiel ist der Code 1 mit dem String „Sendung“ verknüpft. Aus Übersichtlichkeitsgründen wurde in der grafischen Darstellung der Objekte in dieser Diplomarbeit jedoch auf den Verweis verzichtet. Das heißt, daß an dieser Stelle der endgültige Wert angegeben wird (siehe Abb. 3.4 Knoten *Ref*).

3.2.3 Abweichungen zu FESAD im BR-Modell

Aus der Analyse einer AXF-Beispieldatei und einer XSLT-Datei, die die Transformation eines FESAD-Datenexportes nach AXF und damit ins BR-Datenmodell beschreibt, wurde sichtbar, daß das BR-Datenmodell einige Unterschiede zur FESAD-Spezifikation aufweist. Unter anderem wurden 46 weitere Attribute, die aus einem sog. Automationsdatensatz stammen, in die Tabelle *DU* eingefügt. Weitere Unterschiede werden in diesem Kapitel erörtert.

3.2.3.1 Hierarchieebenen

Die Hierarchische Zusammensetzung von *DU*-Objekten wurde im BR-CMS nicht mittels der FESAD-Tabelle *CONTAINER* realisiert. In AXF übernimmt ein Element namens *Link*, das auf *DU*-Elemente verweist und die Art der Referenzierung angibt (*isChildOf*, *isParentOf*, *hasVersionOf*, *isVersionOf*) diese Aufgabe (siehe Abb. 3.5). Jedes *MAObject*

beinhaltet einen eindeutigen Identifier GUID¹⁰, auf den in zu *Link* untergeordneten Knoten referenziert wird. In Abb. 3.5 besitzt eine *DU* mit dem GUID „1234“ eine hierarchisch untergeordnete *DU* mit dem GUID „9876“, die jedoch nicht in der gleichen AXF-Datei eingebunden ist.

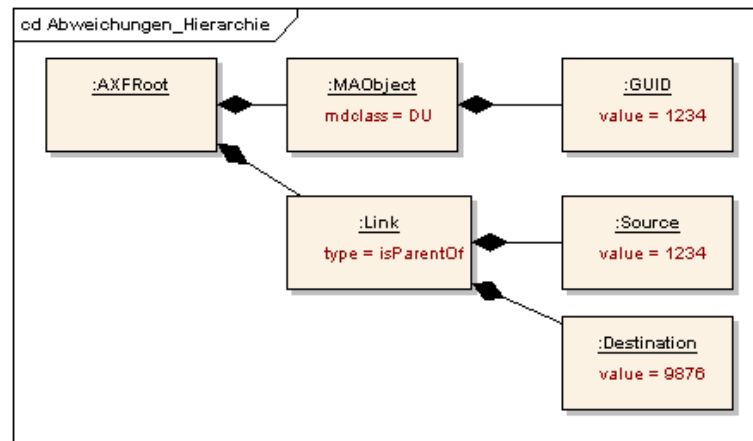


Abbildung 3.5: Hierarchieabbildungen in AXF

3.2.3.2 Straten

Im BR-Datenmodell ist es möglich, Metadaten-Sequenzen (sog. Straten) zu erstellen. Diese Sequenzen enthalten Metadaten zu zeitlich definierten Abschnitten einer *DU*. So können z.B. einzelne Bildabschnitte inhaltlich separat beschrieben werden. Insgesamt gibt es sechs Typen von Straten. Drei beziehen sich auf Bild-, drei auf O-Toninhalte. Je ein Stratum ermöglicht die textuelle Beschreibung eines Bild- bzw. Tonabschnitts. Die übrigen Straten werden auch als Rechtesequenzen bezeichnet und bieten einen Ampelwert, der Aussage über mögliche Verwendungsbeschränkungen gibt (z.B. „Grün“ für freie Verwendung), sowie eine textuelle Aussage zu der Verwendungsbeschränkung (z.B. „keine Freigabe möglich“).

Die eigentlichen BR-Rechteinformationen befinden sich jedoch in einem abgekoppelten Lizenzinformationssystem.

Abb. 3.6 stellt eine AXF-Instanz mit einem Bildinhalt-Stratum dar. Der Hauptknoten

¹⁰GUID: **G**lobally **U**nique **I**dentifier, 128bit-Identifikator, der, wenn gemäß dem Standard ISO/IEC 9834-8:2005 erzeugt, als weltweit eindeutig innerhalb aller GUIDs angesehen werden kann

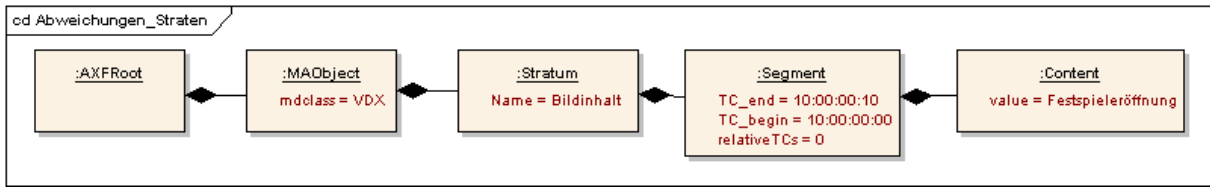


Abbildung 3.6: Straten in AXF

AXFRoot enthält wie immer bei Straten ein *MAObject* vom Typ (*mdclass*= „*VDX*“), in dem das eigentliche *Stratum* untergebracht ist. Dieses *Stratum* enthält beliebig viele Elemente vom Typ *Segment*, die je ein Element vom Typ *Content* besitzen, in dem der Bildinhalt textuell beschrieben werden kann. Das *Segment* macht Angaben zur Zeitspanne (*TC-begin*, *TC-end*), auf die sich die Beschreibung bezieht. Das Attribut *RelativeTCs* gibt an, ob die Timecode-Werte absolut oder relativ zur Timeline der *DU* angegeben wurden.

3.2.3.3 Order

Ein Objekt vom Typ *Order* repräsentiert im BR eine Bestellung von Videomaterial. Der recherchierende Redakteur kann Objekte vom Typ *DU* für eine Weiterverwertung auswählen, die ihm zu geg. Zeitpunkt bereitgestellt werden sollen. Des weiteren kann ein erster Rohschnitt erfolgen, d.h. durch Sichtung der „Low-Resolution“-Kopien können die benötigten Abschnitte grob ausgewählt werden. Eine EDL¹¹ der Schnitte wird erstellt, die dann in NLE¹²-Systemen weiterverwendet werden kann.

Eine *Order* besteht damit aus Objekten vom Typ *DU* oder EDL und besitzt der Bestellung zugeordnete allgemeine Daten, sog. Kopfdaten. Diese Daten können z.B. die Kontaktinformationen des Bestellers umfassen.

Objekte vom Typ *Order* können z.Zt. noch nicht über AXF übertragen werden, liegen daher nur im CMS (BR-Datenmodell) vor.

¹¹EDL: **E**dit **D**ecision **L**ist, framegenaue Schnittliste mit Timecodes und schnittbezogenen Daten

¹²NLE: **N**on-**L**inear **E**editing, computerbasierte Schnittsysteme

3.3 Schnittstellenarchitektur

Sollen Daten aus dem CMS über die in dieser Diplomarbeit entworfene Schnittstelle extrahiert werden, so werden diese zunächst mittels der Zugriffsmethoden in Kap. 6.1 angefordert und CMS-intern über CORBA¹³ verarbeitet. Ein Webservice stellt im nächsten Schritt eine XML-Datei in AXF-Struktur zur Verfügung, die als SOAP¹⁴-Nachricht transportiert wird (siehe Abb. 3.7).

Die Struktur dieser Nachricht soll in der hier zu entwickelnden Schnittstelle zu BMF transformiert werden. Anwendungen, die eine BMF-Schnittstelle besitzen, erhalten so eine dem BMF-Standard entsprechende XML-Datei mit den aus dem CMS extrahierten Informationen.

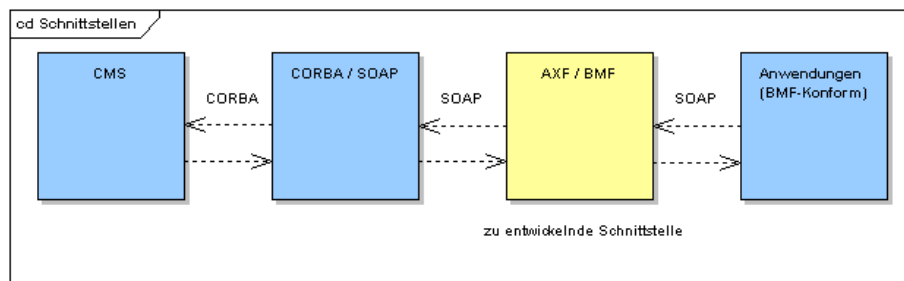


Abbildung 3.7: Schnittstellen im CMS des BR

¹³CORBA: **C**ommon **O**bject **R**quest **B**roker **A**rchitecture, Ansatz zur Entwicklung verteilter heterogener Objektsysteme, standardisiert von der Object Management Group (OMG). [3]

¹⁴SOAP: **S**imple **O**bject **A**ccess **P**rotocol, Protokoll zum Austausch strukturierter Information in dezentralisierten verteilten Umgebungen. Basiert auf XML und Internetprotokollen. [3]

Kapitel 4

Broadcast Metadata exchange Format (BMF)

Das vorliegende Kapitel enthält Informationen zu BMF, dem Broadcast Metadata exchange Format, das vom IRT in Zusammenarbeit mit relevanten Industriepartnern entwickelt wurde. Zunächst wird die Entwicklung des Formates skizziert und grundlegende Konzepte erläutert. Im Anschluß daran werden die für diese Arbeit relevanten Pakete vorgestellt.

4.1 Entwicklung

BMF wurde zur Schaffung einer einheitlichen, standardisierten Struktur von Metadaten entwickelt. Ziel dieser Entwicklung war und ist es, sich von proprietären Insellösungen der einzelnen Rundfunkanstalten zu lösen und einen problemlosen Metadaten austausch zwischen Produktionshäusern und unter Verwendung unterschiedlicher Applikationen zu ermöglichen. Gleichzeitig sollten die schon vorhandenen „in-house“ Metadaten ohne Abstriche in BMF überführt werden können.

Zu Beginn der Entwicklung stand die systematische Analyse der Arbeitsabläufe verschiedener Rundfunkanstalten der ARD. Dabei wurden die Produktionsformen News, Magazin und Feature untersucht. Ein Teilergebnis dieser Analyse, die vereinfachte Darstellung von

Eine weitere wichtige Anforderung an die BMF-Entwicklung war es, problemlosen Datenaustausch zum Datenbankmodell FESAD sicherzustellen, das in allen öffentlichen Rundfunkanstalten (mit Ausnahme des WDR) implementiert wurde. Diese Kompatibilität wird in vorliegender Diplomarbeit untersucht.

Weitere Grundlagen für BMF bilden das „Regelwerk Fernsehproduktion³“, das Pflichtheft für Video-Filetransfer, sowie eine Liste von relevanten Metadaten der EBU⁴ Projektgruppe P/AGTR⁵. [4, S. 5]

Die Umsetzung des BMF-Datenmodells in ein objektorientiertes Klassenmodell ist der vorerst letzte Entwicklungsschritt von BMF und steht kurz vor der Fertigstellung im IRT. Es bietet den Vorteil, das Entitätenmodell um Vererbungsbeziehungen sowie Kompositions- und Aggregationsbeziehungen zu ergänzen. Dies ist u.a. die Voraussetzung zur Integration von BMF als Descriptive Metadata Scheme (DMS) in MXF.

Grundlage dieser Diplomarbeit ist das BMF-Klassenmodell vom 15.08.06.

4.2 Aufbau

Das folgende Kapitel gibt Auskunft über grundlegende Konzepte von BMF. Nähere Informationen finden sich in [5].

4.2.1 Redaktioneller und technischer Teil

Grundsätzlich lässt sich BMF in zwei Teile gliedern:

- der **redaktionelle Teil**, im Modell vor allem durch die Klasse *Abstract_Shot* und die von ihr abgeleiteten Klassen repräsentiert, macht Angaben zu deskriptiven Metadaten (siehe Kap. 2.1) eines Produktionselementes. Als Produktionselemente werden Rohmaterial, Szene, Beitrag und Sendung bezeichnet.

³Regelwerk Fernsehproduktion, Metadaten für den Austausch (Empfehlung der Fernsehbetriebsleiter-Konferenz (FSBL-K)), 03/05

⁴EBU: **E**uropean **B**roadcasting **U**nion

⁵P/AGTR: **A**dvisory **G**roup on **T**apeless **R**ecording, Arbeitsgruppe der EBU

- der **technische Teil**, im Modell vor allem durch die Klasse *ContentInstance* oder zu deutsch „Physikalische Realisierung und Speicherung“ repräsentiert, macht Angaben zu den technischen Metadaten (siehe Kap. 2.1) eines Produktionselementes.

Hierbei ist zu beachten, daß in BMF der redaktionelle Teil zu Programmplanungszwecken auch ohne technischen Teil existieren kann, der technische Teil jedoch nicht ohne den redaktionellen.

4.2.2 Hierarchieebenen

Zur näheren Beschreibung von Programmbestandteilen hat sich die hierarchische Beschreibung als vorteilhaft erwiesen. Sie konnte teilweise aus FESAD übernommen werden. Dabei werden die Produktionselemente **Rohmaterial**, **Szene**, **Beitrag** und **Sendung** unterschieden, die wiederum weitere Elemente enthalten können. Ein Produktionselement Sendung kann somit verschiedene Beiträge beinhalten, die sich wiederum aus Szenen oder direkt aus Rohmaterial zusammensetzen können (siehe Abb. 4.2).

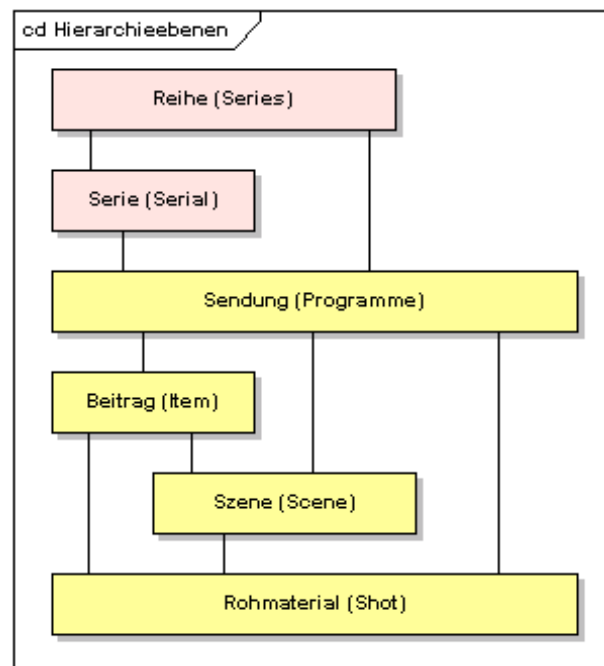


Abbildung 4.2: Hierarchische Struktur zur Beschreibung von Produktionselementen

Mithilfe dieser Elemente lassen sich die Strukturen der o.g. Produktionsformen News, Magazin und Feature (als szenisches Element) nachbilden.

Auch Sendungsgruppierungen werden durch die hierarchische Aufteilung möglich. Dabei unterscheidet BMF zwischen Serien, Reihen und Staffeln.

- **Serien** kennzeichnen sich dadurch, daß die Handlung einer Folge auf der Handlung der vorhergehenden Folge aufbaut.
- **Reihen** kennzeichnen sich dadurch, daß die Handlung eines Elements in sich abgeschlossen ist. Reihen können sowohl einzelne Sendungen als auch Serien beinhalten.
- **Staffeln** kennzeichnen sich dadurch, daß sie Elemente enthalten, die für Serien oder Reihen gemeinsam produziert wurden.

Eine abschnittsweise Beschreibung von Programmbestandteilen geschieht durch sog. **Sequenzen**. Sie ermöglichen eine auf jedes Signal abgestimmte Zuordnung von deskriptiven Metadaten, z.B. können für einzelne Bildsequenzen die darin mitwirkenden Schauspieler angegeben werden.

4.3 Klassenmodell

Die bereits angesprochene Modellierung des Datenmodells BMF als objektorientiertes Klassenmodell ist die Grundlage für das Mapping der Daten zwischen dem existierenden Datenmodell des BR und BMF. Daher liefert dieses Kapitel eine kompakte Einführung in das BMF-Klassenmodell mit Stand vom 15.08.06.

Tab. 4.1 gibt einen Überblick über die aktuelle BMF-Paketstruktur. Die für diese Arbeit relevanten Pakete werden im Folgenden nähergehend beschrieben. Ausführliche Informationen zu allen Paketen finden sich in [5].

BMF-konforme Produkte müssen das Klassenmodell nicht in seiner Gesamtheit implementieren; es genügen die für den jeweiligen Anwendungsfall notwendigen Teile des Modells. Beispielsweise ist für die redaktionelle Programmplanung eine Implementierung des technischen Sendeplans nicht zwingend erforderlich.

Die Pakete *Participant*, *Right*, *Award*, *Category*, *MainDescriptor* und Teile vom Paket *Sequence* wurden im Rahmen dieser Diplomarbeit entworfen.

| Paket | Untergeordnete Pakete | Funktion |
|------------|--|--|
| Programme | ProductionElements Grouping Participant Right Programmplanung Schemaplanung Script | <i>Programme</i> ist das Herzstück der BMF-Architektur und beschreibt Geschäftsobjekte, die auch Produktionselemente genannt werden. |
| Production | ContentInstance Planning PEState FileTransfer LineExchange Language | <i>Production</i> beschreibt produktionsrelevante Daten sowie technische Metadaten. |
| Publishing | Transmission Internet Releases | <i>Publishing</i> beschreibt die Publikation von Produktionselementen. |
| Annotation | AdditionalInformation Award Category MainDescriptor Sequence | <i>Annotation</i> ermöglicht es, beschreibende Metadaten zu Produktionselementen anzugeben. |

Tabelle 4.1: Die Paketstruktur von BMF

4.3.1 Paket Programme

4.3.1.1 Productionelements

Im Paket *Productionelements* finden sich die als Produktionselemente bezeichneten Klassen *Programme* (Sendung), *Item* (Beitrag), *Scene* (Szene) und *Shot* (Rohmaterial), die Auskunft über redaktionelle Zusammensetzungen geben.

Diese Produktionselemente sind durch eine Vererbungshierarchie miteinander verbunden (siehe Abb. 4.3). *Abstract_Shot* ist die Basisklasse, von der sich alle anderen Produktionselemente ableiten. Sie bieten daher dieselben Attribute und Methoden wie *Abstract_Shot* an und ergänzen diese um die für sie spezifischen Eigenschaften.

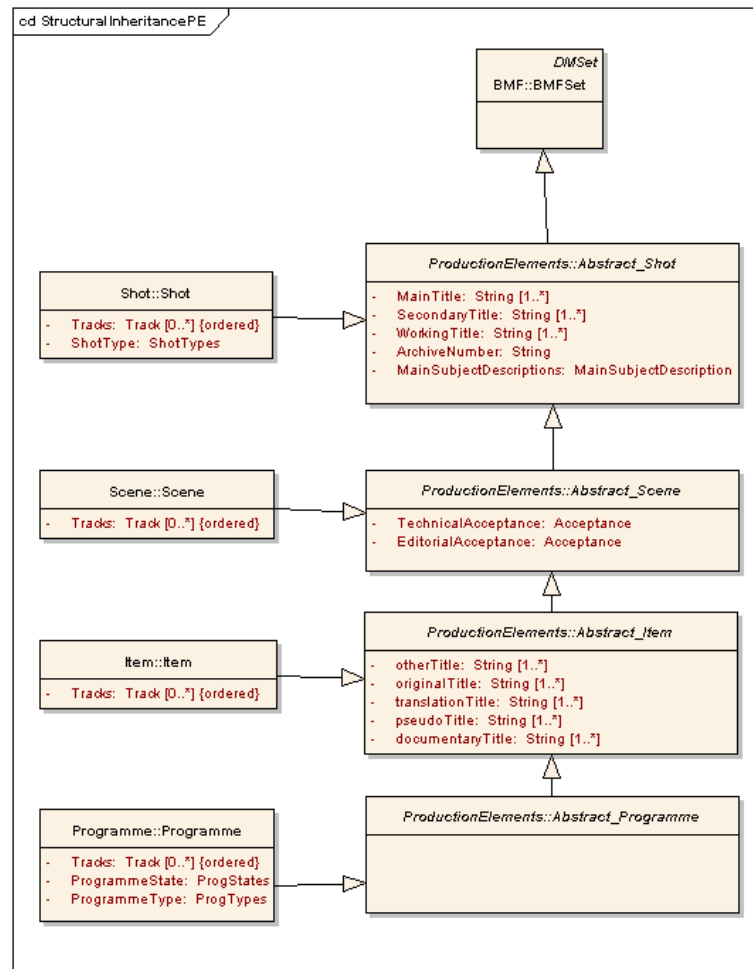


Abbildung 4.3: Vererbungshierarchie der Produktionselemente

Produktionselemente besitzen deskriptive Metadaten als Attribute, wie z.B. Titel, Arbeitstitel und Angaben zum Sachinhalt.

Um Hierarchieebenen unter Produktionselementen darzustellen, besitzen die Elemente einen Array von Elementen vom Typ *EssenceTrack* (siehe Abb. 4.4). Diese Tracks repräsentieren ein einzelnes Signal (z.B. eine Audiospur), besitzen eine eigene Identifikationsnummer (*TrackID*) und genau eine *EssenceSequence*. Dieses Konstrukt der (1:1)-Beziehung wurde aus Kompatibilitätsgründen zu MXF übernommen. Die *EssenceSe-*

quence definiert Dauer des Signals (*Duration*), Art des Signals (*DataDefinition*) und referenziert eine bis beliebig viele von der abstrakten Klasse *EssenceSourceReference* abgeleitete Klassen. Diese referenzieren auf untergeordnete Produktionselemente und deren Elemente vom Typ *EssenceTrack* und beinhalten die Attribute Startzeitpunkt (*DestinationStartPosition*), Dauer des Beitragsausschnitts (*DestinationDuration*) sowie dessen Platzierung im übergeordneten Produktionselement (*StartPosition*, *Duration*).

Abb. 4.4 zeigt beispielhaft den Aufbau eines Produktionselementes *Programme*, das sich aus einem Produktionselement *Item* zusammensetzt.

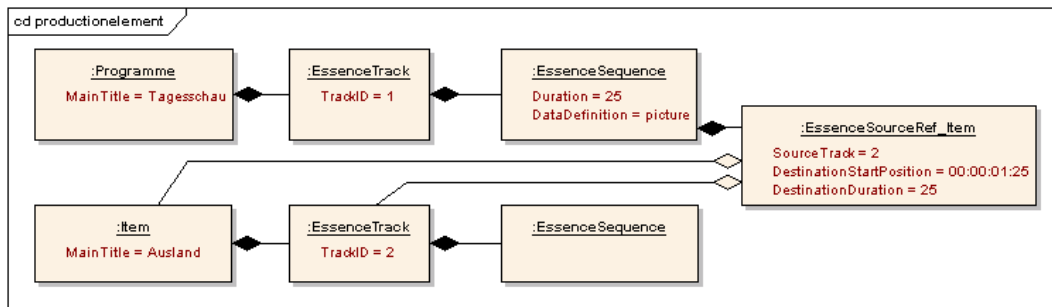


Abbildung 4.4: Struktureller Aufbau einer Sendung mit einem Beitrag

4.3.1.2 Grouping

Wie bereits erwähnt, bietet BMF die Möglichkeit, Sendungsgruppierungen abzubilden. Unterscheidungsmerkmale der im Paket *Grouping* definierten Gruppierungen *Serial* (Serie), *Series* (Reihe) und *LineUp* (Staffel) sind im Abschnitt 4.2.2 einzusehen.

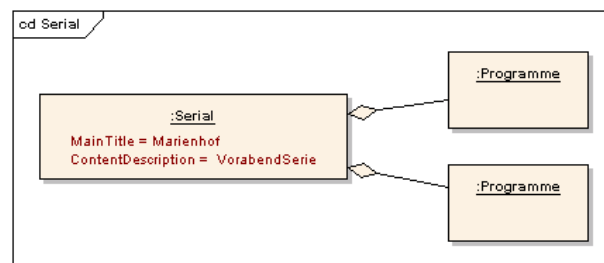


Abbildung 4.5: Struktureller Aufbau einer Serie

Abb. 4.5 stellt eine Serie mit zwei zugehörigen Programmelementen dar.

Basisklasse von *Serial* und *Series* ist die Klasse *Grouping*, die allgemeine Attribute wie

Titel der Gruppierung (*MainTitle*), Inhaltsbeschreibung (*ContentDescription*) oder eine Referenz der Klasse *Branding*, die Vermarktungsinformationen mitführt, besitzt. Die abgeleitete Klasse *Serial* besitzt einen Array für Elemente vom Typ *Programme* und Attribute, die die Anzahl der durchnummerierten Sendungselemente angeben.

4.3.1.3 Participant

Im Paket *Participant* können Mitwirkende einer Produktion mit ihren Kontaktdaten ausgewiesen und eine Liste der für die Produktion relevanten Orte (*LocationList*) angelegt werden.

Eine Kontaktliste (*ContactList*) ermöglicht den direkten Zugriff auf alle Beteiligten einer Produktion. Eine Teilnehmerliste (*ParticipationList*) gibt Auskunft über die Tätigkeit eines Beteiligten. Diese Trennung ist sinnvoll, wenn beispielsweise ein Mitwirkender mehrere Funktionen ausfüllt (siehe Abb. 4.6).

Die Klasse *ContactList* enthält einen Array vom Typ *LegalPerson* (dt. juristische Person), von der sich Objekte vom Typ *Organisation* und *IndividualPerson* (dt. natürliche Person) ableiten lassen. Jede juristische Person enthält ein Objekt vom Typ *Address*, in der Angaben zu Kontaktdaten der Person oder Organisation gemacht werden können.

Um allen Beteiligten Arbeitsbereiche zuordnen zu können, gibt es die Klasse *ParticipationList*, die einen Array vom Typ der abstrakten Klasse *Participant* besitzt. Von *Participant* abgeleitete Klassen sind *Creator* (dt. Urheber), *ProductionStaff* (dt. Produktion), *Contributors* (dt. Mitwirkende) und *Others* (dt. Sonstige). Sie enthalten jeweils eine eigene Enumerationlist, die eine Auswahl von vorgegebenen Tätigkeitsbeschreibungen bietet. Jeder *Participant* enthält eine Weak-Reference auf die Klasse *LegalPerson*. Damit kann jede natürliche Person mehrere Tätigkeitsbereiche erfüllen. Die Zugehörigkeit der *Participant*-Objekte zur *ParticipationList* wurde mittels Weak-References gelöst, da die Klasse *Participant* von *DMDData* abgeleitet ist und daher *DMSegment* eine Strong-Reference auf sie besitzt (siehe Abb. 4.13).

Abb. 4.6 stellt eine mitwirkende Organisation und natürliche Person dar, denen Tätigkeiten zugeordnet wurden. Zudem wurde ein Produktionsort angegeben.

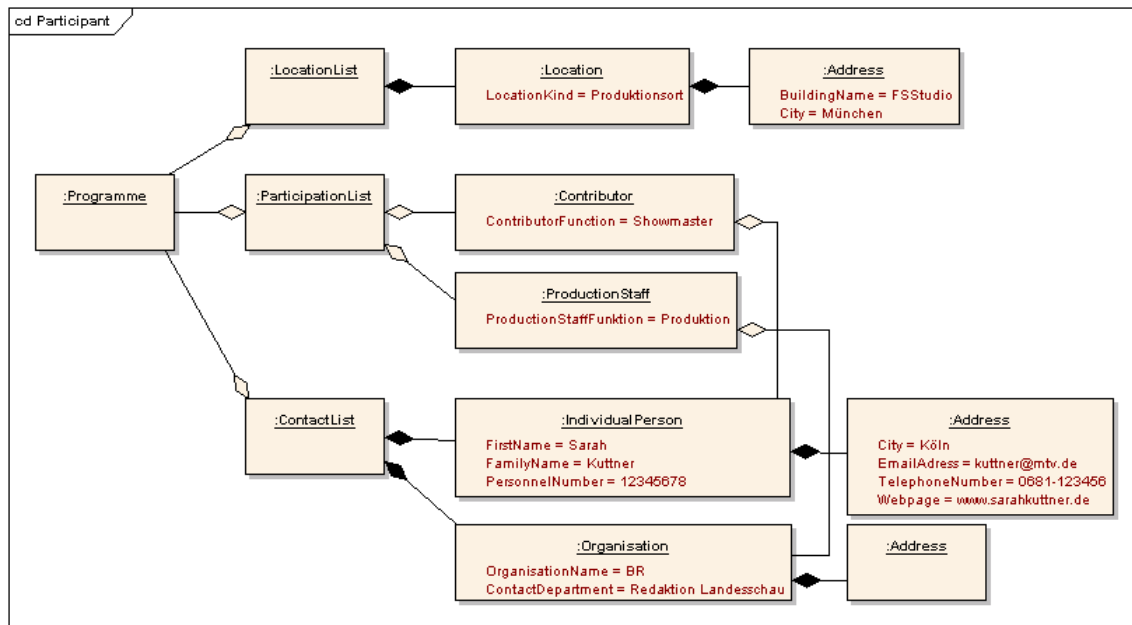


Abbildung 4.6: Darstellung zwei Mitwirkender und eines Produktionsortes

4.3.1.4 Right

Die Klasse *Rights* ist über eine *RightsList* an das Produktionselement angeschlossen und ermöglicht die Angabe von Rechteinformationen (siehe Abb. 4.7).

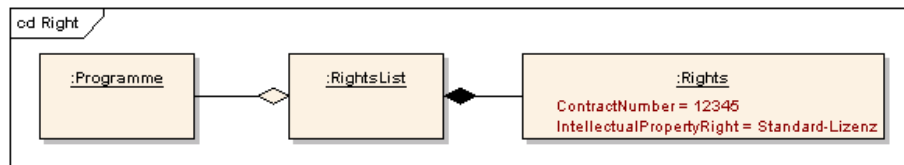


Abbildung 4.7: Angabe eines Rechtes für eine Sendung

4.3.1.5 Programmplanung

Das Paket *Programmplanung* spezifiziert Programmplanung und ist korrespondierend zum Paket *Transmission* (siehe Kap. 4.3.3.1) anzusehen.

Die Klasse *BroadcastChannel* repräsentiert einen Sender, der beliebig viele Elemente vom Typ *ProgrammeSchedule* (dt. Programmplan) enthält. Im Regelfall wird pro Ausstrahlungstag ein *ProgrammeSchedule* erstellt, der sich aus beliebig vielen Elementen vom Typ *ScheduleElement* (dt. Programmelement) zusammensetzt. Die Klasse *ScheduleElement*

repräsentiert einen „Sende-Abschnitt“ und enthält die für den Programmplatz relevanten Daten wie beispielsweise ungefähre Dauer der Ausstrahlung (*TransmissionDuration*), Sendeplatzbezeichnung (*Schema*) oder Ausstrahlungstitel (*TransmissionTitle*). Sie referenziert weiterhin genau ein Element der Klasse *Programme*. Produktionselemente anderen Typs können nicht gesendet werden. Des weiteren besitzt das Paket eine oder keine Klasse vom Typ *TransmissionSchedule* (dt. Sendeplan), der Angaben zur technischen Sendeplanung macht. Nähere Erläuterungen hierzu im Paket *Transmission* (siehe Kap. 4.3.3.1).

Abb. 4.8 stellt die Einbindung einer Sendung in den Programmplan des Hessischen Rundfunks zur Ausstrahlung am 31.01.08 dar.

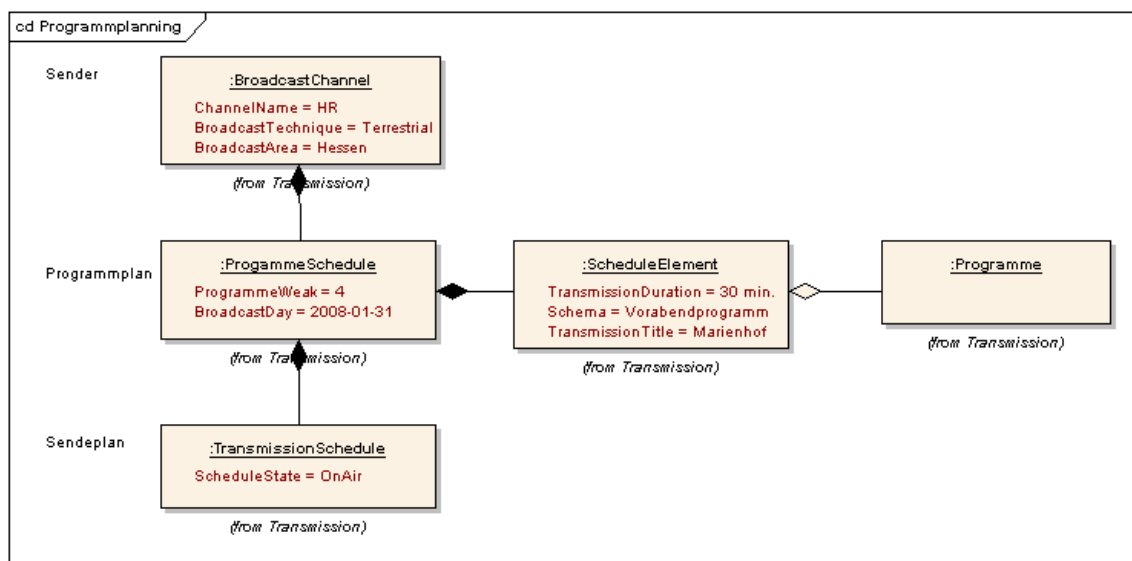


Abbildung 4.8: Darstellung eines Programmplanes

4.3.2 Paket Production

4.3.2.1 ContentInstance

Im Paket *ContentInstance* sind die technischen Metadaten der Essenz enthalten. Da die technischen Metadaten jedoch nicht primärer Bestandteil des Mappings dieser Arbeit sind, sondern nur vereinzelt (siehe Kap. 6.2.1.2) in Erscheinung treten, wird das Paket hier nur in aller Kürze beschrieben. Für nähere Informationen bezüglich der Struktur der

ContentInstance wird auf [5] verwiesen.

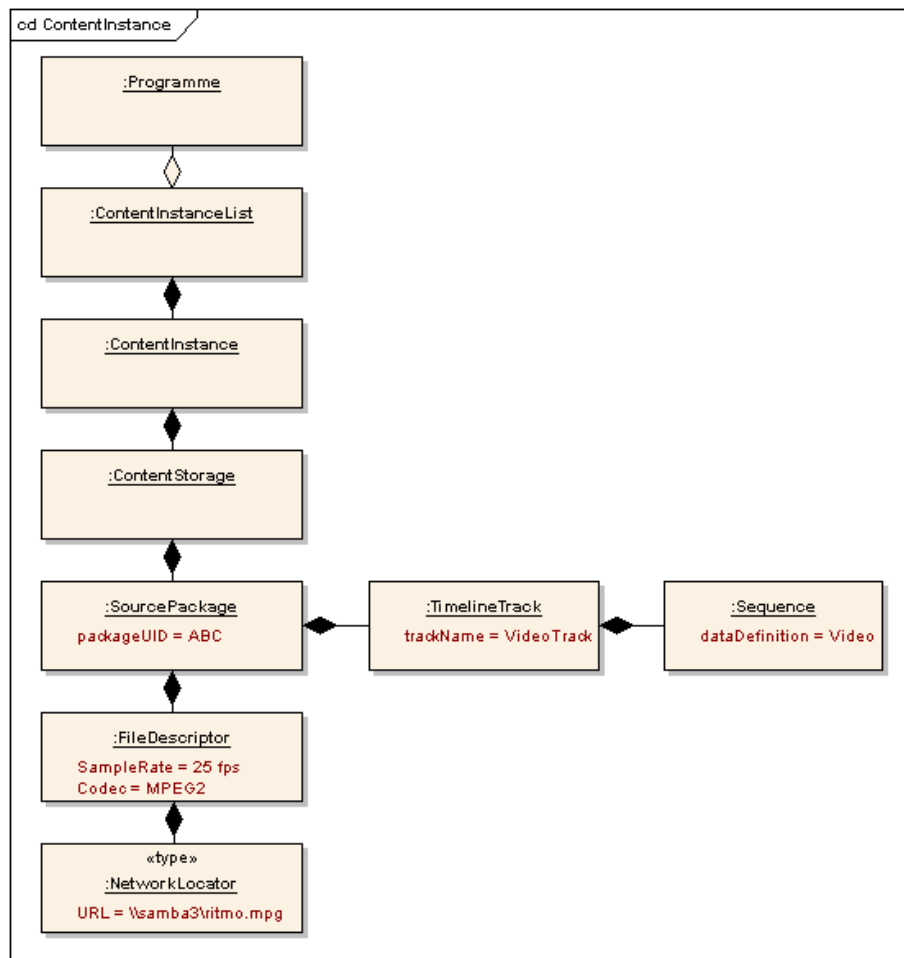


Abbildung 4.9: Abbildung der Essenz für ein Produktionselement

Abb. 4.9 stellt die Abbildung von Essenz im Paket *ContentInstance* dar. Übergeordnetes Objekt ist eine *ContentInstanceList*, die auf ein Produktionselement verweist und eine Liste von *ContentInstance*-Objekten enthält. Jedes dieser Objekte repräsentiert eine Essenz zu einem bestimmten Zeitpunkt und Ort. Dieses Zusammenspiel zwischen den Produktionselementen (also von *Abstract_Shot* abgeleiteten Klassen) und der *ContentInstance* ermöglicht es, mehrere Essenz-Instanzen eines Produktionselements darzustellen. Beispielsweise kann ein und derselbe Film sowohl auf Festplatte als auch auf Magnetband vorliegen. Abgebildet würde dies durch ein Produktionselement, das zwei Objekte vom Typ *ContentInstance* besitzt.

Eine *ContentInstance* besitzt immer genau einen *ContentStorage*, der als Container für

von der abstrakten Klasse *GenericPackage* abgeleiteten Objekte (*MaterialPackage* oder *SourcePackage*) fungiert. Diese Objekte ermöglichen es, die segmentierte Speicherung von Essenz darzustellen. Befindet sich ein Beitrag z.B. auf zwei unterschiedlichen Bändern oder Dateien, so wird dies entsprechend durch zwei Objekte vom Typ *SourcePackage* dargestellt.

Ein *SourcePackage* besitzt immer einen oder mehrere Deskriptoren, also Objekte vom Typ der abstrakten Klasse *GenericDescriptor*, die Angaben zur physikalischen Speicherung der Essenz machen. Ein in *GenericDescriptor* eingebundenes Objekt vom Typ *Locator* spezifiziert dann letztendlich den Speicherort der Essenz.

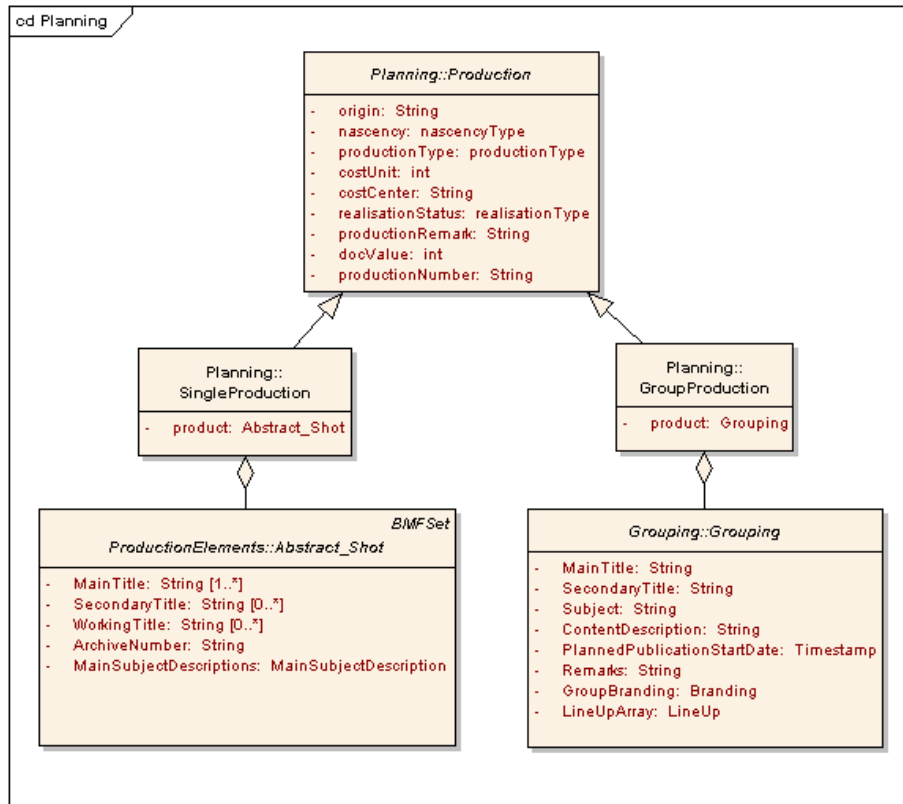
Ein *SourcePackage* besitzt weiterhin (1..n) Signale, also von der abstrakten Basisklasse *GenericTrack* abgeleitete Klassen. Dieser Aufbau orientiert sich am Track-Mechanismus des Pakets *Productionelements* (siehe Kap. 4.3.1.1).

In Abb. 4.9 besitzt ein Produktionselement vom Typ *Programme* eine *ContentInstance*, die ein Videosignal und dessen Speicherort auf einem Server beschreibt.

4.3.2.2 Planning

Das Paket *Planning* ermöglicht die Disposition und Planung produktionsbezogener Ressourcen. Für diese Diplomarbeit ist lediglich ein kleiner Ausschnitt des Paketes relevant. Produktionen lassen sich in BMF in *SingleProduction* (dt. Einzelproduktion) und *GroupProduction* (dt. Gruppenproduktion) einteilen. Gemeinsame Basisklasse ist die abstrakte Klasse *Production*, die produktionsrelevante Daten wie Kostenträger (*CostCenter*), Realisationsstatus (*realisationStatus*) oder die Produktionsnummer (*productionNumber*) enthält. Die Klasse *SingleProduction* verweist auf genau ein Produktionselement, für das Produktionsdaten gültig sind, die Klasse *GroupingProduction* auf genau eine Gruppierung, für die Produktionsdaten gelten.

Abb. 4.10 stellt die Vererbungshierarchie von Produktionen und deren Referenzierung auf Produktionselemente bzw. Elemente vom Typ *Grouping* im Klassenmodell dar.

Abbildung 4.10: Vererbungshierarchie im Paket *Planning*

4.3.2.3 PESTate

Das Paket *PEState* ermöglicht es, den Produktionselementen *Sequence*, *Item* und *Programme* einen definierten Status zuzuweisen. Dieser Status kann folgende Werte annehmen: geplant (*PEPlanned*), angemeldet (*PERequested*), genehmigt (*PEApproved*), Rohschnittliste liegt vor (*PERoughEditList*), in Produktion (*PEInProduction*), produziert (*PEProduced*) und annulliert (*PECancelled*). Des weiteren gibt es die Stati inhaltliche Abnahme (*EditorialAcceptance*) und dokumentarische Erfassung (*ArchiveAnnotation*), die genauer spezifiziert werden können.

Im Klassenmodell sind alle Stati von der abstrakten Klasse *PEState* abgeleitet. Ein Produktionselement besitzt immer genau einen Status als Attribut. Die Klassen *EditorialAcceptance* und *ArchiveAnnotation* besitzen als Attribut eine *EnumerationList*, die es erlaubt, den jeweiligen Status weiter auszudifferenzieren.

Abb. 4.11 stellt ein Produktionselement vom Typ *Szene* dar, das den Status der inhaltli-

chen Abnahme besitzt, der wiederum auf „akzeptiert“ (*accepted*) gesetzt wurde.

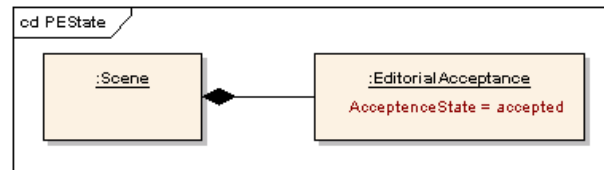


Abbildung 4.11: Ein Produktionselement *Scene*, das inhaltlich genehmigt wurde

4.3.3 Paket Publishing

4.3.3.1 Transmission

Das Paket *Transmission* spezifiziert die Sendeplanung und ist korrespondierend zum Paket *Programmplanung* (siehe 4.3.1.5) anzusehen.

Im Gegensatz zum Paket *Programmplanung* werden den Elementen vom Typ *Programme* im Paket *Transmission* ihre für die Sendung notwendigen technischen Daten, die *ContentInstance*, zugeordnet (siehe Abb. 4.12).

Die Klasse *TransmissionSchedule* bildet den redaktionellen Programmplan technisch ab. Je nach ihrem Status (*ScheduleState*), der auf *Scheduled*, *OnAir* oder *AsRun* gesetzt sein kann, ergibt sich die Bedeutung des *TransmissionSchedule*. Der Status *Scheduled* bezeichnet die vorläufige technische Planung eines Programmplans, die von der tatsächlichen Sendung abweichen kann. Der Status *OnAir* spezifiziert die tatsächlich erfolgte Ausstrahlung der in ihr referenzierten Elemente vom Typ *TransmissionScheduleData*. Der Status *AsRun* fungiert als Protokollierung des automatisierten Playouts.

Die Klasse *TransmissionSchedule* enthält korrespondierend zur Klasse *ProgrammeSchedule* beliebig viele Elemente vom Typ *TransmissionScheduleElement*, die die zu den Objekten vom Typ *Programme* zugehörigen Elemente vom Typ *ContentInstance* einbindet. Die Klassen *TransmissionScheduleElement* (Paket *Transmission*) und *ScheduleElement* (Paket *Programmplanung*) leiten sich von der abstrakten Klasse *ScheduleAbstractElement* ab.

Die Klasse *TransmissionScheduleData* ermöglicht segmentierte Einbindung von Elementen

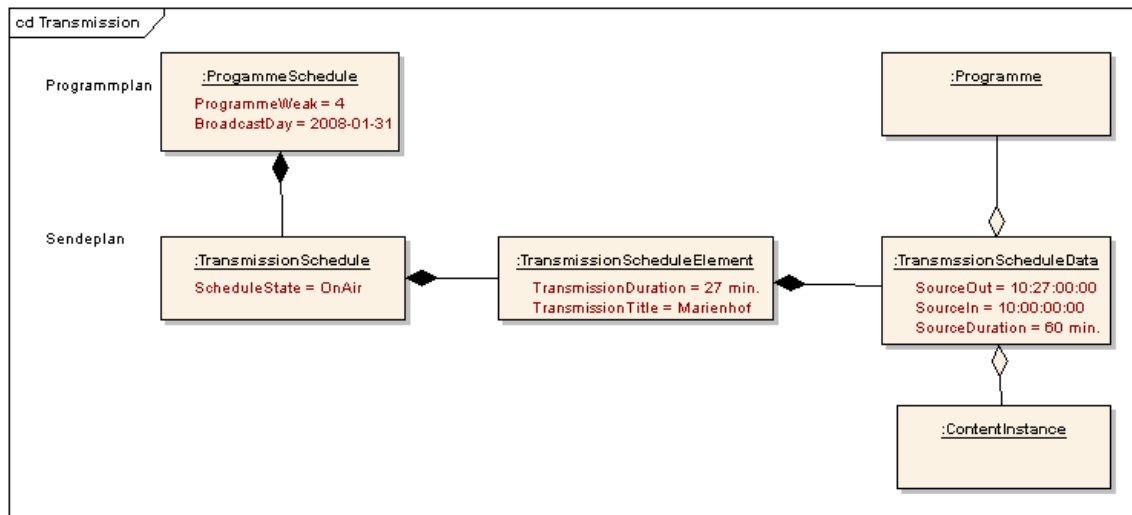


Abbildung 4.12: Darstellung eines Sendeplans

ten vom Typ *ContentInstance* und lässt die Angabe von Start-, Endzeitpunkten (*SourceIn*, *SourceOut*) und Dauer (*Duration*) zu.

Abb. 4.12 stellt korrespondierend zu Abb. 4.8 die Sendeplanung dar.

4.3.4 Paket Annotation

4.3.4.1 Sequence

Das Paket *Sequence* ermöglicht es, einzelne Abschnitte und Signale (Elemente vom Typ *EssenceTrack*) eines Produktionselements näher zu beschreiben. Im Gegensatz zu den direkten Attributen eines Produktionselements gelten die hier verwendeten Metadaten nicht notwendigerweise für die Gesamtheit eines Produktionselements.

Abb. 4.13 stellt eine Sendung dar, die in *DMSegment* auf eine Klasse *ImageDescription* referenziert, die als Bildbeschreibung fungiert.

Der Aufbau einer *Sequence* ist am redaktionellen Aufbau von Produktionselementen (siehe Kap. 4.3.1.1) orientiert und gliedert sich wie folgt: eine *DMTrackList*⁶ referenziert das zugehörige Produktionselement und dient als Container für alle vorhandenen Elemente vom Typ *DMTrack*. Ein *DMTrack* repräsentiert ein Signal für Metadaten, besitzt eine

⁶DMTrackList: Descriptive Metadata Track List

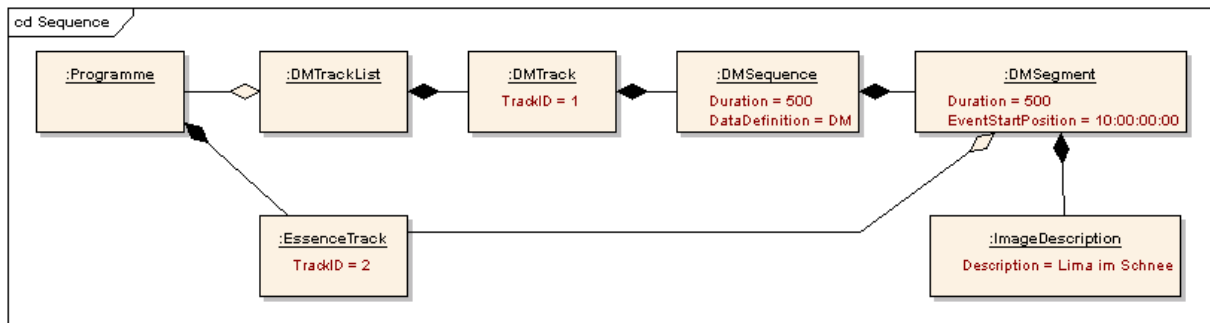


Abbildung 4.13: Angabe einer Bildbeschreibung für eine Sendung

eindeutige *TrackID* und ist über eine (1:1)-Beziehung mit einer *DMSequence* verbunden. Dieses Konstrukt wurde aus Kompatibilitätsgründen zu MXF übernommen. Eine *DMSequence* definiert die Art der referenzierten Daten, im Paket *Sequence* immer *DM* („Descriptive Metadata“), und besitzt einen Array von Elementen vom Typ *DMSegment*. Ein *DMSegment* beinhaltet genau ein Objekt vom Typ der abstrakten Klasse *DMDData*, der Basisklasse für deskriptive Metadaten.

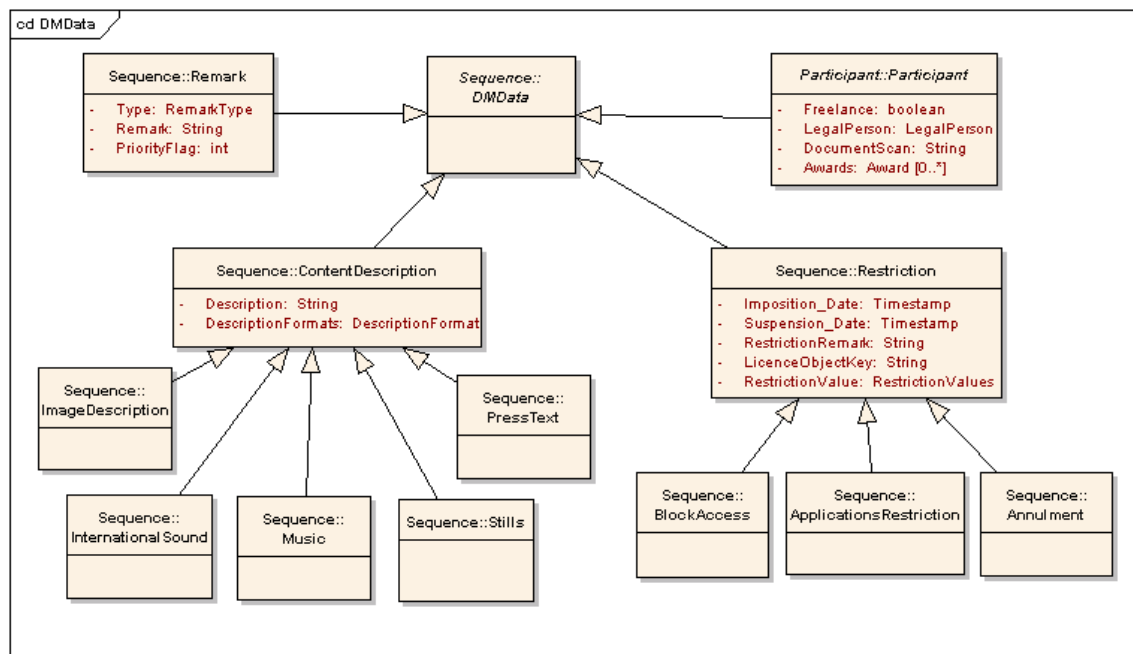
Abb. 4.14 stellt die Vererbungshierarchie von deskriptiven Metadaten dar, die in *DMSegment* eingebunden werden können. Hierzu zählen u.a. Anmerkungen (*Remark*), Beschreibungen (*ContentDescription*), Beschränkungen (*Restriction*) und Teilnehmer (*Participant*).

Weiterhin definiert *DMSegment* Zeitabschnitt, Dauer und Position der eingebundenen deskriptiven Metadaten. Um diese der Essenz eindeutig zuzuordnen, referenziert *DMSegment* auf beliebig viele Elemente vom Typ *EssenceTrack*, für die die Metadaten gelten können.

Ein Sonderfall tritt ein, wenn die in einer *Sequence* angegebenen Metadaten für die gesamte Dauer des Produktionselementes gelten sollen. In diesem Fall kann die Dauer (*Duration*) in *DMSegment* auf „null“ gesetzt werden.

4.3.4.2 AdditionalInformation

Das Paket *AdditionalInformation* ermöglicht es, Zusatzinformationen zu Produktionselementen anzugeben. Diese können in Binärform im Datenmodell oder als externer Link

Abbildung 4.14: Vererbungshierarchie von *DMDData*

vorliegen. Realisiert wurde dies durch eine abstrakte Klasse *AdditionalInformation*, von der die Klassen *BinaryInformation* und *ExternalInformation* abgeleitet wurden. Die Klasse *AdditionalInformationList* fungiert als eine Art Container, in dem sich alle Zusatzinformationen befinden.

Das Attribut *mimeType* der Klasse *BinaryInformation* spezifiziert das Format der Binärdaten (z.B. application/PDF).

Abb. 4.15 stellt binär und als Referenz vorliegende Zusatzinformationen dar.

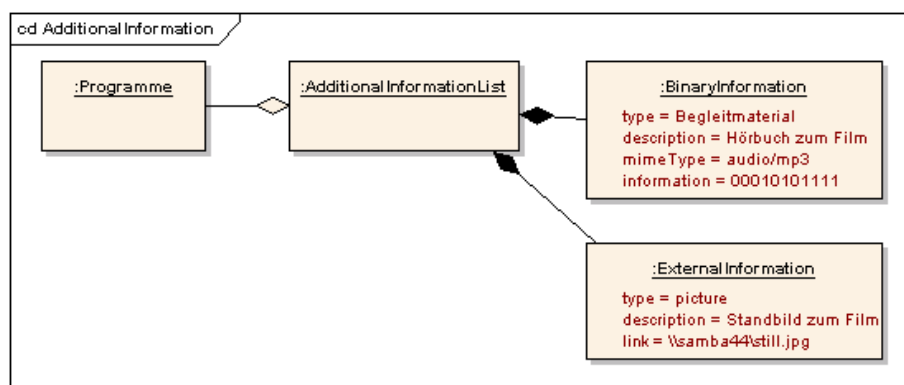


Abbildung 4.15: Angabe von Zusatzinformationen zu einer Sendung

4.3.4.3 Award

Im Paket *Award* können Produktionselementen und Teilnehmern einer Produktion Auszeichnungen bei Wettbewerben zugewiesen werden. Eine *AwardList* ermöglicht den Zugriff auf alle Auszeichnungen, die dieses Produktionselement betreffen. Der *Award* selbst besitzt zusätzlich eine (n:n)-Beziehung zu Mitwirkenden, die als Gewinner einer Auszeichnung gelten.

In Abb. 4.16 wird einem Mitwirkenden einer Produktion eine Auszeichnung zugewiesen.

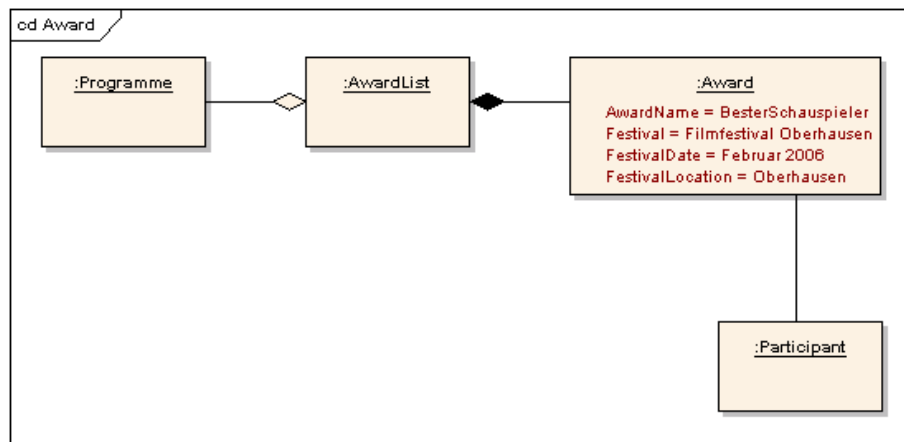


Abbildung 4.16: Darstellung eines auf einem Festival ausgezeichneten Mitwirkenden

4.3.4.4 Category

Das Paket *Category* erlaubt es, Produktionselemente in den Kategorien Inhalt (*Content*), Präsentationsform (*PresentationForm*), Zielgruppe (*TargetGroup*), Verwendung (*IntendedPurpose*), Sparte (*Section*) und Eignung (*Suitability*) einzuteilen. Die Klasse *CategoryList* fungiert als Container für von der abstrakten Klasse *Category* abgeleiteten Klassen. Jedes der abgeleiteten Elemente besitzt eine EnumerationList *CategoryType*, die alle vorgegebenen Kategoriewerte enthält.

In Abb. 4.17 wird der Kategorie Präsentationsform der Wert *magazine* (dt. Magazin) zugewiesen.

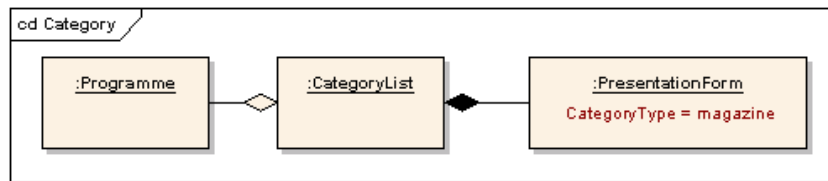


Abbildung 4.17: Zuweisung einer Kategorie für eine Sendung

4.3.4.5 MainDescriptor

Das Paket *MainDescriptor* ermöglicht die Verschlagwortung eines Produktionselementes in den Kategorien Allgemein (*General*), Themen (*Subjects*), Orte (*Locations*), Personen (*Persons*) und Zeit (*Age*). Eine *MainDescriptorList* enthält alle von der abstrakten Klasse *MainDescriptor* abgeleiteten Klassen.

Abb. 4.18 stellt die Verschlagwortung der Kategorie Allgemein dar.

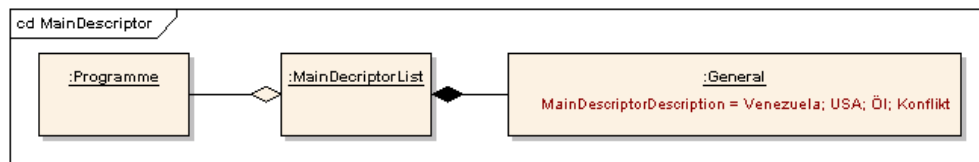


Abbildung 4.18: Verschlagwortung einer Sendung

Kapitel 5

Anforderungen an die Schnittstelle

Um den Entwurf der Schnittstelle an objektive Kriterien zu koppeln, wurden vom BR Anforderungen gestellt, die wie folgt zusammengefasst werden können:

1. Es sollen alle für den Austausch relevanten Informationen des BR-Datenmodells auf das BMF-Modell abgebildet werden.

Die für den Austausch relevanten Informationen finden sich in den FESAD-Tabellen *DU*, *ID_OF_DU*, *BC*, *RAPC*, *TITLE*, *APC*, *CONTENT*, *CATEGORY*, *MD*, *INFO*, *LINK*, *REMARK*, *RESTRICTION*, *LICENCE* sowie *CONTAINER*.

2. Eine Dokumentarische Einheit (*DU*, siehe Kap. 3.2.1.1) soll mitsamt aller Datensätze anderer Tabellen, die Informationen über sie enthalten, extrahiert werden können. Dazu zählt auch die hierarchische Zusammensetzung dieser bis zur ersten Hierarchieebene. Das heißt, daß die zu der *DU* in Beziehung stehenden weiteren Elemente vom Typ *DU* ebenfalls zurückgegeben werden sollen. Die Datensätze anderer Tabellen, die diese Elemente der ersten Hierarchieebene beschreiben, sollen je nach Wunsch mit zurückgegeben werden oder nicht.

Die *DU* soll durch Angabe eines weltweit eindeutigen Identifiers ausgewählt werden können.

3. Eine Liste aller Rechtesequenzen einer *DU* soll extrahiert werden können. Eine Rechtesequenz ist ein Stratum (siehe Kap. 3.2.3.2), das zu einem zeitlich definierten Abschnitt einer Essenz Rechteinformationen zur Verfügung stellt.

Die Rechtesequenzen sollen durch Angabe eines weltweit eindeutigen Identifiers ausgewählt werden können, der sich auf die *DU* bezieht, in der sich die Sequenzen befinden.

4. Alle in einer *Order* (siehe Kap. 3.2.3.3) befindlichen Elemente (Objekte vom Typ *DU* oder EDL) sollen samt der *Order*-Kopfdaten extrahiert werden können. Dazu zählt auch die hierarchische Zusammensetzung der Objekte vom Typ *DU* bis zur ersten Hierarchieebene. Das heißt, daß die zu einer *DU* in Beziehung stehenden weiteren Elemente vom Typ *DU* ebenfalls zurückgegeben werden sollen. Die Datensätze anderer Tabellen, die diese Elemente der ersten Hierarchieebene beschreiben, sollen je nach Wunsch mit zurückgegeben werden oder nicht.

Die *Order* soll durch Angabe eines weltweit eindeutigen Identifiers ausgewählt werden können.

Kapitel 6

Entwurf der Schnittstelle

Basierend auf den im vorigen Kapitel genannten Anforderungen kann die Schnittstelle nun entworfen werden. Zunächst werden Zugriffsmethoden mit Übergabeparametern und Rückgabewerten definiert. Die Transformation der ausgegebenen Daten in das BMF-Modell schließt das Kapitel und damit den Entwurf der Schnittstelle ab.

Da der in Kapitel 3.2 durchgeführte Vergleich des BR-Datenmodells mit FESAD zeigte, daß es Unterschiede sowohl in der Struktur als auch im Informationsgehalt der Modelle gibt, werden an entsprechenden Stellen der Datentransformation beide Modelle betrachtet.

6.1 Zugriffsmethoden

Die folgenden Zugriffsmethoden dieses Kapitels erfüllen die zuvor gestellten Anforderungen und besitzen Rückgabewerte, deren Struktur BMF-konform ist. Der Informationsumfang entspricht den aus dem CMS des BR extrahierten Daten, die in AXF-Struktur vorliegen.

6.1.1 getPEFull

`getPEFull(Identifizier) : PEFull`

Die Methode *getPEFull*¹ dient dazu, Informationen zu *DU*-Objekten zu identifizieren und auszugeben. Umgesetzt werden die Anforderungen 1 und 2 des vorangegangenen Kapitels.

Übergabeparameter der Methode ist ein eindeutiger Identifier vom Typ UUID oder GUID², der die *DU* selektiert.

Rückgabewert der Methode ist ein Objekt vom Typ *PEFull* (siehe Kap. 6.2.1).

6.1.2 getPEStandard

`getPEStandard(Identifizier) : PEStandard`

Die Methode *getPEStandard* dient dazu, Informationen zu *DU*-Objekten zu identifizieren und auszugeben. Umgesetzt werden die Anforderungen 1 und 2 des vorangegangenen Kapitels. Die Methode deckt sich im Wesentlichen mit *getPEFull*, überträgt jedoch weniger Informationen der hierarchisch verknüpften Produktionselemente.

Übergabeparameter der Methode ist ein eindeutiger Identifier vom Typ UUID oder GUID, der die *DU* selektiert.

Rückgabewert der Methode ist ein Objekt vom Typ *PEStandard* (siehe Kap. 6.2.1).

6.1.3 getRightsequence

`getRightsequence(Identifizier) : DMSegment[]`

Die Methode *getRightsequence* dient der Rückgabe der Informationen, die in allen Rechtefolgen einer *DU* enthalten sind. Umgesetzt wird Anforderung 3 des vorangegangenen Kapitels.

¹PE: **P**roduction**E**lement

²GUID: **G**lobally **U**nique **I**dentifier und UUID: **U**niversally **U**nique **I**dentifier, 128bit-Identifikatoren, die, wenn gemäß dem Standard ISO/IEC 9834-8:2005 erzeugt, als weltweit eindeutig innerhalb aller GUIDs bzw. UUIDs angesehen werden können

Übergabeparameter ist ein eindeutiger Identifier vom Typ `UUID` oder `GUID`, der die *DU* bestimmt, auf die sich die Rechtesequenzen beziehen.

Rückgabewert ist ein Array vom Typ *DMSegment*. *DMSegment* besitzt eine Strong-Reference auf Objekte vom Typ *Restriction* und eine Weak-Reference auf betreffende Elemente vom Typ *EssenceTrack* (siehe Kap. 4.3.4.1).

6.1.4 getFullOrder

`getFullOrder(Identifier) : FullOrder`

Die Methode *getFullOrder* dient der Rückgabe der Informationen, die in einer *Order* enthalten sind. Umgesetzt wird Anforderung 4 des vorangegangenen Kapitels.

Übergabeparameter ist ein eindeutiger Identifier vom Typ `UUID` oder `GUID`, der die *Order* bestimmt, in der Elemente vom Typ *DU* oder EDL enthalten sind.

Rückgabewert ist ein Objekt vom Typ *FullOrder* (siehe Kap. 6.2.3).

6.1.5 getStandardOrder

`getStandardOrder(Identifier) : StandardOrder`

Die Methode *getStandardOrder* dient der Rückgabe der Informationen, die in einer *Order* enthalten sind. Umgesetzt wird Anforderung 4 des vorangegangenen Kapitels. Die Methode deckt sich im Wesentlichen mit *getFullOrder*, überträgt jedoch weniger Informationen, da hier nur Verweise auf die hierarchisch verknüpften Produktionselemente von Objekten vom Typ *DU* zurückgegeben werden.

Übergabeparameter ist ein eindeutiger Identifier vom Typ `UUID` oder `GUID`, der die *Order* bestimmt, in der Elemente vom Typ *DU* oder EDL enthalten sind.

Rückgabewert ist ein Objekt vom Typ *StandardOrder* (siehe Kap. 6.2.3).

6.2 Datentransformation

Aufgrund der nun bekannten relevanten Übergabeparameter und Rückgabewerte der Zugriffsmethoden kann das Mapping dieser Rückgabewerte erfolgen. In FESAD und AXF vorliegende Informationen werden in der hier zu entwickelnden Schnittstelle nach BMF transformiert. Das XML-Austauschformat AXF repräsentiert hierbei stellvertretend das BR-Datenmodell mit seinen Eigenheiten, das wie erwähnt nicht veröffentlicht ist.

Eine detaillierte Dokumentation der FESAD-, AXF- und BMF-Attribute findet sich in [8], [7] und in [5].

Die Unterüberschriften dieses Kapitels orientieren sich an den Klassennamen des FESAD-Modells. Abbildungen dienen zur prinzipiellen Darstellung der Transformation und verwenden daher im Regelfall FESAD-, AXF- und BMF-Beispielinstanzen.

ERD-spezifische Attribute wie z.B. Sortierreihenfolge oder Fremdschlüssel sind für das Mapping nicht von Bedeutung und werden daher nicht aufgeführt. In der Regel werden materialspezifische Attribute abgebildet. Prozessbezogene Attribute, wie beispielsweise der Status von Sichtungskopien, finden keine Abbildung, da sie interne Arbeitsabläufe spezifizieren und für den BMF-Austausch nicht von Bedeutung sind.

Das Mapping der Klassenvariablen wird tabellarisch dargestellt. Im Regelfall genügt es dort, die Datenmodelle FESAD und BMF aufzuzeigen, wenn der Informationsgehalt des BR-Modells (in Abbildungen durch AXF-Instanzen repräsentiert) mit dem von FESAD identisch ist. In Kap. 6.2.2 wird AXF jedoch ergänzend hinzugenommen, da dort der grundsätzlich unterschiedliche Mechanismus der Straten (siehe Kap. 3.2.3.2) angewendet wird. In Kap. 6.2.3 werden das BR-Datenmodell und BMF tabellarisch gegenüber gestellt, da die Objekte vom Typ *Order* z.Zt. in AXF noch nicht übertragen werden können und in FESAD nicht vorhanden sind.

Typographische Konvention

Alle im Mapping verwendeten Variablen werden in UML-Notation dargestellt, um die exakte Klassenhierarchie zu berücksichtigen (Variable:: Variable:: usw.).



Abbildung 6.1: Beispiel zur Klassennotation des Mappings

Abb. 6.1 zeigt beispielhaft ein Klassendiagramm, indem die Klasse *SinglePEFull* ein Recht *Rights* über eine *RightsList* besitzt. Die *RightsList* dient dabei als Container für eine beliebige Anzahl von Rechten. Will man nun den Ort des Attributs *ContractNumber* angeben, so würde dies wie folgt geschehen: ***rights:: Rights[]:: ContractNumber***. Der erste Wert (im Beispiel: *rights*) bezieht sich in unserem Fall auf die Klasse *SinglePEFull* und ist vom Typ *RightsList*. In der Klasse *RightsList* gibt es wiederum einen Array namens *Rights[]* vom Typ *Rights*. Alle Elemente vom Typ *Rights* enthalten das Attribut namens *ContractNumber*.

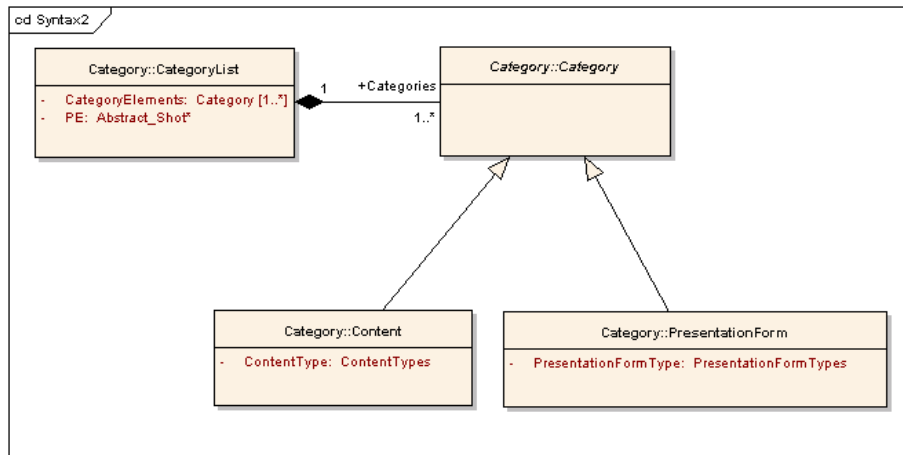


Abbildung 6.2: Sonderfall zur Klassennotation des Mappings

Zu berücksichtigen ist der Sonderfall, daß die Klasse, in der sich das Attribut befindet, durch die Notation nicht zweifelsfrei angegeben werden kann. Siehe hierzu das Klassendiagramm in Abb. 6.2. Die Variable *ContentType* befindet sich im Beispiel nur in der

abgeleiteten Klasse *Content*. Daher wäre die Angabe ***CategoryElements[]:: Content-Type[]*** nicht eindeutig, weshalb folgende Notation gewählt wurde: ***CategoryElements[] (Content):: ContentType[]***

Runde Klammern geben also im Zweifelsfall die Klasse des Attributes an. Wird eine Variable auf mehrere Variablen abgebildet, so sind diese durch Kommata getrennt.

Alle BMF-Klassenpfade beziehen sich auf die Rückgabewerte der jeweiligen Zugriffsmethoden. Das heißt, daß sich die Klassenpfade in Kap. 6.2.1 auf die Klassen *SinglePEFull* und *SinglePEStandard*, in Kap. 6.2.2 auf *DMSegment* und in Kap. 6.2.3 auf *FullOrder* und *StandardOrder* beziehen.

6.2.1 Abbildung DU auf SinglePEFull / SinglePEStandard

Die für den Rückgabewert der Funktionen *getPEFull* und *getPEStandard* relevante *DU* wird in BMF auf Objekte vom Typ *SinglePEFull* bzw. *SinglePEStandard* abgebildet, die wiederum in Objekten vom Typ *PEFull* und *PEStandard* enthalten sind.

Die Klasse *SinglePEFull* unterscheidet sich von *SinglePEStandard* darin, daß sie die Informationen aller FESAD-Tabellen enthält, die sich auf die Tabelle *DU* beziehen, während *SinglePEStandard* nur Informationen der Tabelle *DU* selbst enthält (siehe Abb. 6.3).

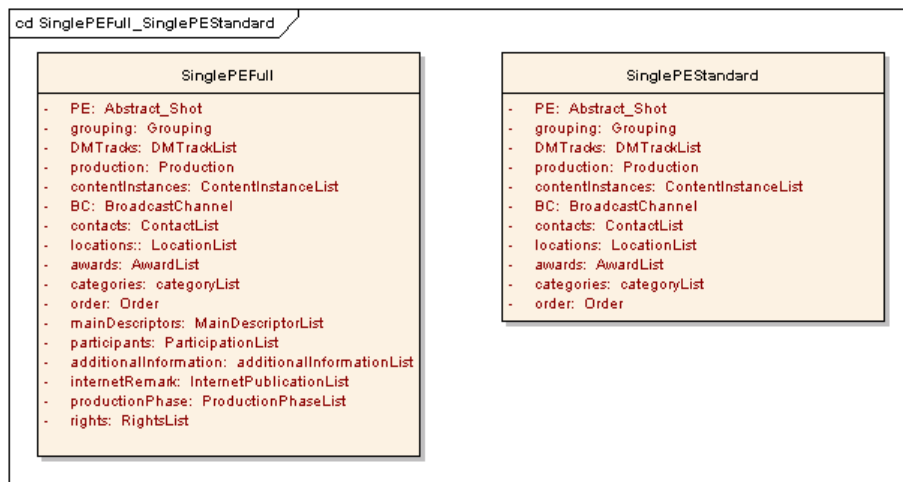
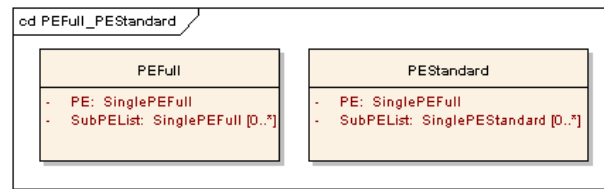


Abbildung 6.3: Die Klassen *SinglePEFull* und *SinglePEStandard*

Diese expliziten Klassenerstellungen sind u.a. deshalb notwendig, weil viele der Klassen, die sich auf ein Produktionselement beziehen, umgekehrt Weak-referenziert sind und daher nicht automatisch mitgeliefert werden (siehe Kap. 2.3). Für sie fungieren *SinglePEFull* und *SinglePEStandard* als eine Art Container.

Die Rückgabewerte der Funktionen *getPEFull* und *getPEStandard* sind die Klassen *PEFull* und *PEStandard* (siehe Abb. 6.4). Sie beinhalten Informationen zur angeforderten *DU* (*PE*) und einen Array von hierarchisch untergeordneten Produktionselementen (*SubPEList*). Dies ist notwendig, da in dem angeforderten Produktionselement *PE* lediglich Weak-Referenzen auf hierarchisch untergeordnete Produktionselemente bestehen (siehe Kap. 4.3.1.1). Der Unterschied zwischen den Klassen *PEStandard* und *PEFull* ist, daß die Elemente der *SubPEList* unterschiedlichen Types sind.

Abbildung 6.4: Die Klassen *PEFull* und *PEStandard*

Alle in diesem Kapitel im Mapping angegebenen BMF-Attribute beziehen sich auf die Klassen *SinglePEFull* und *SinglePEStandard*.

6.2.1.1 Hierarchische Abbildung (CONTAINER, Link)

Während in FESAD die hierarchische Zusammensetzung einer *DU* über eine Tabelle namens *CONTAINER* geregelt ist, wurde beim Datenmodell des BR eine Tabelle namens *Link* eingeführt. In BMF übernehmen die von *EssenceSourceReference* abgeleiteten Klassen diese Aufgabe. Zur Verdeutlichung sei in Abb. 6.5 eine Sendung mit Beitrag in FESAD, AXF und BMF dargestellt.

Zum Mapping der FESAD-Tabelle *CONTAINER*:

| FESAD | BMF | Bemerkungen |
|----------------|--|---|
| PARENT_KEY | | Elternelement ist immer das Produktionselement, welches die von <i>EssenceSourceReference</i> abgeleitete Klasse beinhaltet |
| PARENT_STATION | | siehe <i>PARENT_KEY</i> |
| CHILD_KEY | | Kindelement ist immer das Produktionselement, das in von <i>EssenceSourceReference</i> abgeleiteten Klassen referenziert wird |
| CHILD_STATION | | siehe <i>CHILD_KEY</i> |
| TC | PE:: Tracks[]:: Sequence:: StructuralComponents[]:: DestinationStartPosition | |
| DURATION | PE:: Tracks[]:: Sequence:: StructuralComponents[]:: DestinationDuration | |

Fortsetzung

| | | |
|-------|-----|--|
| FESAD | BMF | Bemerkungen |
| TAKE | - | Redundante Angabe, da sich die Reihenfolge der eingebundenen Produktionselemente über den <i>TC</i> -Wert ergibt |

Tabelle 6.1: Hierarchische Abbildungen

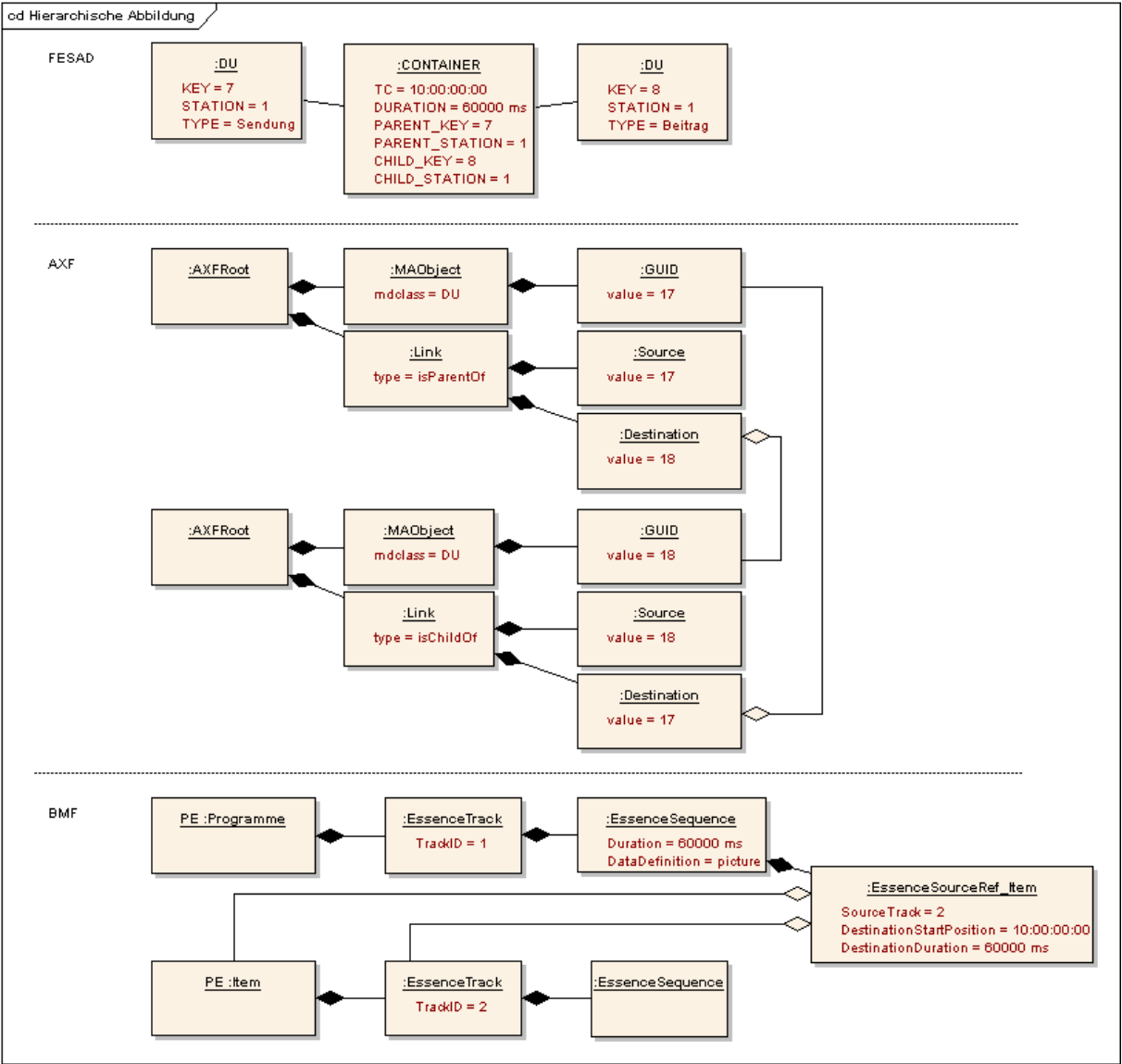


Abbildung 6.5: Hierarchische Abbildung einer *DU*-Instanz auf eine BMF-Instanz

Es ist im AXF-Modell nicht möglich, Timecode-Werte zur segmentierten Einbindung von Elementen vom Typ *DU* anzugeben. Eine hierarchisch untergeordnete *DU* muß immer in voller Länge verwendet werden.

6.2.1.2 Dokumentarische Einheit (DU, Automationsdatensatz)

Die FESAD-Tabelle *DU* wird in BMF in diversen Paketen abgebildet. Im BR-Datenmodell enthält die Tabelle neben den in FESAD spezifizierten Attributen den BR-spezifischen Automationsdatensatz (siehe Kap. 3.2.3) als (1:1)-Beziehung.

Das Mapping der FESAD-Tabelle *DU* findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|------------------|--|--|
| KEY | PE:: Archividentifier | Wird gemeinsam mit <i>STATION</i> abgebildet |
| STATION | PE:: Archividentifier | Wird gemeinsam mit <i>KEY</i> abgebildet |
| ID | production:: productionNumber | Wird beim BR als Produktionsnummer verwendet. |
| TYPE | | Abbildung erfolgt durch unterschiedliche Produktionselemente, d.h. von <i>Abstract_Shot</i> abgeleitete Klassen |
| POOL | - | Keine Rundfunk-übergreifenden Pools, daher nicht abgebildet |
| AWARDS | awards:: AwardList[]:: AwardName, awards:: AwardList[]:: FestivalDate | Informationen werden getrennt abgebildet |
| ORIGIN | contacts:: LegalPersons[] (Organisation):: OrganisationName | Wird als <i>Organisation</i> abgebildet, auf die ein Element vom Typ <i>ProductionStaff</i> verweist, das in seinem Attribut <i>ProductionStaffList</i> den Wert <i>producer</i> gesetzt hat |
| LOCATION | locations:: Locations[]:: LocationDescription | Die Variable <i>LocationKind</i> der selben Klasse wird auf <i>Drehort</i> gesetzt |
| NASCENCY | production:: NascencyType | |
| PROD.PROCESS | production:: ProductionType | |
| PROD.PROCESS_SUP | production:: ProductionRemark | |
| DOC.VALUE | categories:: CategoryElements[] (Suitability):: DocumentaryLevel | |
| COST_UNIT | - | Redundante Angabe, da bereits <i>Cost_Center</i> und Produktionsnummer angegeben sind |
| COST_CENTER | production:: CostCenter | |

Fortsetzung

| FESAD | BMF | Bemerkungen |
|---------------|---|---|
| REALISATION | production:: ProductionState | |
| STATUS | PE:: PESTates (ArchiveAnnotation):: ArchiveAnnotation | Die Variable <i>PEStates</i> gibt es nur in von <i>Abstract_Scene</i> abgeleiteten Klassen. |
| STATUS_DRA | - | Für Austausch nicht relevant |
| PLANNED_USE | - | Wird beim BR nicht verwendet |
| REQUEST_DATE | - | Für Austausch nicht relevant |
| IS_DELETED | - | |
| DELETION_DATE | - | |
| DELETION_TEXT | - | |
| CREATION_DATE | - | |
| CORR_DATE | - | |
| PROC | - | |

Tabelle 6.2: Abbildung der Tabelle *DU*

Das Mapping des Automationsdatensatzes findet wie folgt statt:

| BR-Modell | BMF | Bemerkungen |
|------------------------|---|--|
| File-ID | PE:: InternalProductionIdentifier | |
| Titel | PE:: MainTitle | |
| Sende-Reihentitel | grouping:: MainTitle | Serien und Reihen leiten sich in BMF von der gemeinsamen Basis-klassse <i>Grouping</i> ab. Daher erfolgt das Mapping auf dessen Attribut <i>Main-Title</i> |
| Start of message (SOM) | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: tracks[] (TimelineTrack):: origin | |
| Duration | PE:: Tracks[]:: Sequence:: Duration | <i>Duration</i> ist Attribut der Klasse <i>Sequence</i> , die es für jeden <i>EssenceTrack</i> eines Produktionselementes gibt |
| File-TC In | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: tracks[]:: sequence:: structuralComponents[] (SourceClip):: startPosition | |

Fortsetzung

| BR-Modell | BMF | Bemerkungen |
|---------------------------|--|---|
| File-TC Out | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: tracks[]:: sequence:: structuralComponents[] (SourceClip):: startPosition + duration | Werte von <i>startPosition</i> und <i>duration</i> werden addiert |
| Duration-File | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: tracks[]:: sequence:: structuralComponents[] (SourceClip):: duration | |
| Materialkategorie | | Siehe Tab. 6.4 |
| Workarea/Folio-ID | - | Befindet sich in BMF aktuell in Entwicklung |
| Tonstatus | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: descriptor (GenericSoundEssenceDescriptor):: electroSpatialFormulation | |
| Tonspurbelegung Spur 1 | - | Befindet sich in BMF aktuell in Entwicklung |
| Tonspurbelegung Spur 2 | - | |
| Tonspurbelegung Spur 3 | - | |
| Tonspurbelegung Spur 4 | - | |
| Bildformat | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: descriptor (GenericPictureEssenceDescriptor):: AspectRatio | |
| Videoformat | - | Uneindeutige Information, daher nicht abgebildet |
| VT-Untertitel | contentInstances:: ContentInstanceList[]:: ContextRelatedInfo | |
| Erstellungsdatum | - | Für Austausch nicht relevant |
| Letzter Sendevorgang | categories:: CategoryElements[] (Suitability):: LastPossibleTransmissionTime | |
| Operator | - | Für Austausch nicht relevant |
| Letzte Worte/Bild | DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[]:: Metadata (CueWordOut):: WordsOut | |

Fortsetzung

| BR-Modell | BMF | Bemerkungen |
|-------------------------------|--|---|
| Allgemeine Bemerkungen | PE:: Remarks:: Remark | |
| Allg. Bemerkung-Datum-Uhrzeit | PE:: Remarks:: RemarkDateAndTime | |
| gBrowsingCopy (Live-Feed) | - | Informationen über interne Prozesse, daher keine Abbildung |
| gBrowsingCopy (Z-Storage) | - | |
| gBrowsingCopy (Archive) | - | |
| BrowsingCopy available | - | |
| BrowsingCopy Status | - | |
| Send to Archive | - | |
| Archived | - | |
| Archive Status | - | |
| Redaktionelle Abnahme | PE:: PEStates (EditorialAcceptance):: Remark | Da Feld erst nach redaktioneller Abnahme belegt ist, kann das Attribut <i>AcceptanceState</i> in <i>EditorialAcceptance</i> auf <i>accepted</i> gesetzt werden |
| Technische Abnahme | contentInstances:: ContentInstanceList[]:: TechnicalAcceptance:: Remark | Da Feld erst nach technischer Abnahme belegt ist, kann das Attribut <i>CIAcceptanceState</i> in <i>TechnicalAcceptance</i> auf <i>ready-for-transmission</i> gesetzt werden |
| Order-ID | - | Ist kein Attribut eines Produktionselementes, sondern einer Bestellung. Daher keine Abbildung |
| RD-Event-ID | - | Befindet sich bei BMF in Entwicklung und ist kein Attribut des Produktionselementes |
| TS-Event-ID | BC:: ProgrammScheduleArray[]:: ProgrammeScheduleElement[]:: InstanceUID | |
| Digitalisierungskennung | contentInstances:: ContentInstanceList[]:: contentStorage:: packages[]:: descriptor (TextLocator):: digitalisationIdentifier | |
| Placeholder Field 5 | - | Wird beim BR nicht verwendet |

Fortsetzung

| BR-Modell | BMF | Bemerkungen |
|---------------------|--|--|
| Format | contentInstances:: ContentInstanceList[]:: ContentStorage:: packages[]:: descriptor (GenericPictureEssenceCoding):: PictureEssenceCoding, contentInstances:: ContentInstanceList[]:: ContentStorage:: packages[]:: descriptor:: Locators | Informationen werden getrennt abgebildet: Kompressionsinformationen werden in <i>PictureEssenceCoding</i> abgebildet. Auskunft über das verwendete Fileformat gibt die Dateierweiterung in <i>NetworkLocator</i> |
| Kostenstelle | production:: costCenter | |
| Produktionsnummer | production:: productionNumber | |
| Besteller/Redakteur | - | Ist kein Attribut eines Produktionselementes, daher keine Abbildung |
| Redaktion | contacts:: LegalPersons:: Departments:: DepartmentName | |
| Monochrom | contentInstances:: ContentInstanceList:: ContextRelatedInfo | |
| Cleanfeed | contentInstances:: ContentInstanceList:: ContextRelatedInfo | |

Tabelle 6.3: Abbildung des BR-Automationsdatensatzes

Das Attribut *Materialkategorie* des Automationsdatensatzes besitzt unterschiedliche Bedeutungsdimensionen, weshalb diese Informationen in BMF an unterschiedlichen Orten abgebildet werden:

| FESAD | BMF | Bemerkungen |
|---|---------------------------------|---|
| Rohmaterial, Beitrag | | Abbildung über Produktionselemente, d.h. von <i>Abstract_Shot</i> abgeleitete Klassen |
| Sendung BR, Sendung FF | | Abbildung über Sendungen und deren Angaben im Paket <i>Participant</i> |
| Vorschnitt | PE:: PEStates (PERoughEditList) | |
| Sponsoring, Vorspann, Abspann, Mitschnitt | PE (Programme):: ProgTypes | |

Fortsetzung

| FESAD | BMF | Bemerkungen |
|-----------------------|------------------------------|------------------------|
| Nachrufe, Promo, Spot | - | Keine Abbildung in BMF |
| EBU-Feed | order:: SupplierOrganisation | |

Tabelle 6.4: Attribut *Materialkategorie*

6.2.1.3 Titel (TITLE)

Die Tabelle *Title* bezeichnet je nach Wert ihrer Variable *TYPE* unterschiedliche Titel, deren Mapping in BMF an unterschiedlichen Stelle erfolgt:

| FESAD | BMF | Bemerkungen |
|---------------------|--|---|
| SerienReihentitel | grouping:: MainTitle | Serien und Reihen leiten sich in BMF von der gemeinsamen Basisklasse <i>Grouping</i> ab. Daher erfolgt das Mapping auf dessen Attribut <i>MainTitle</i> |
| SendeHaupttitel | BC:: ProgrammScheduleArray[]:: ProgrammscheduleElements[]:: TransmissionTitle, PE:: MainTitle | Der Sendetitel eines Produktionselements findet sich in BMF im Paket <i>Programmplanning</i> , der Haupttitel in den von <i>Abstract_Shot</i> abgeleiteten Produktionselementen |
| Untertitel | PE:: SecondaryTitle | |
| Serien-Untertitel | grouping:: SecondaryTitle | |
| Beitrags-Untertitel | PE:: SecondaryTitle | |
| Sonstiger Titel | - | Ist im aktuellen FESAD-Implementationsstand nicht mehr vorhanden |
| Arbeitstitel | PE:: WorkingTitle | |
| Beitragstitel | PE:: MainTitle | |
| Originaltitel | PE:: MainTitle | Der Originaltitel ist der Titel eines Produktionselementes in seiner ersten Version. In BMF werden neue Versionen von Produktionselementen in neuen Instanzen abgebildet. |
| Übersetzungstitel | PE:: TranslationTitle | |
| Vorlagentitel | PE:: ArtWorkOriginalTitle | |
| Quasititel | PE:: PseudoTitle | |

Fortsetzung

| FESAD | BMF | Bemerkungen |
|------------------------|---|---|
| Dokumentarischer Titel | PE:: DocumentaryTitle | |
| Magazintitel | grouping:: MainTitle | In BMF gibt es keine expliziten Magazin-sendungen. Sie werden im Paket <i>Grouping</i> abgebildet |
| Einspieltitel | PE:: IngestTitle | |
| Programmstruktur-Titel | BC:: ProgrammScheduleArray[]:: ProgramScheduleElements[] (ScheduleElement):: Schema | |

Tabelle 6.5: *TYPE*-Werte der Tabelle *TITLE* und deren Auswirkung auf das Mapping

Eine Übersicht über das Mapping aller in der Tabelle *TITLE* vorhandenen Attribute:

| FESAD | BMF | Bemerkungen |
|--------|---|--|
| TYPE | | siehe Tab. 6.5 |
| TITLE | | siehe Tab. 6.5 |
| PART | grouping (Series):: SeriesProgrammeArray, grouping (Serial):: SerialProgrammeArray | Abbildung durch geordnete Auflistung von Programmen innerhalb von Arrays |
| REMARK | - | Keine Abbildung, da Anmerkungen an anderer Stelle im Modell möglich sind |

Tabelle 6.6: Mapping der Tabelle *TITLE*

Das FESAD-Attribut *PART* liefert eine Aussage über die Reihen- und Folgennummer einer *DU*. In BMF werden Reihen und Folgen im Paket *Grouping* abgebildet. Die diesbezügliche Nummer eines Produktionselementes ist durch die geordnete Auflistung von Programmen innerhalb von Arrays erkennbar. Diese Arrays befinden sich in von der Klasse *Grouping* abgeleiteten Klassen (vgl. Kap. 4.3.1.2).

Abb. 6.6 stellt die Abbildung eines Titels vom Typ *SendeHaupttitel* dar. In BMF werden Sende- und Haupttitel separat abgebildet.

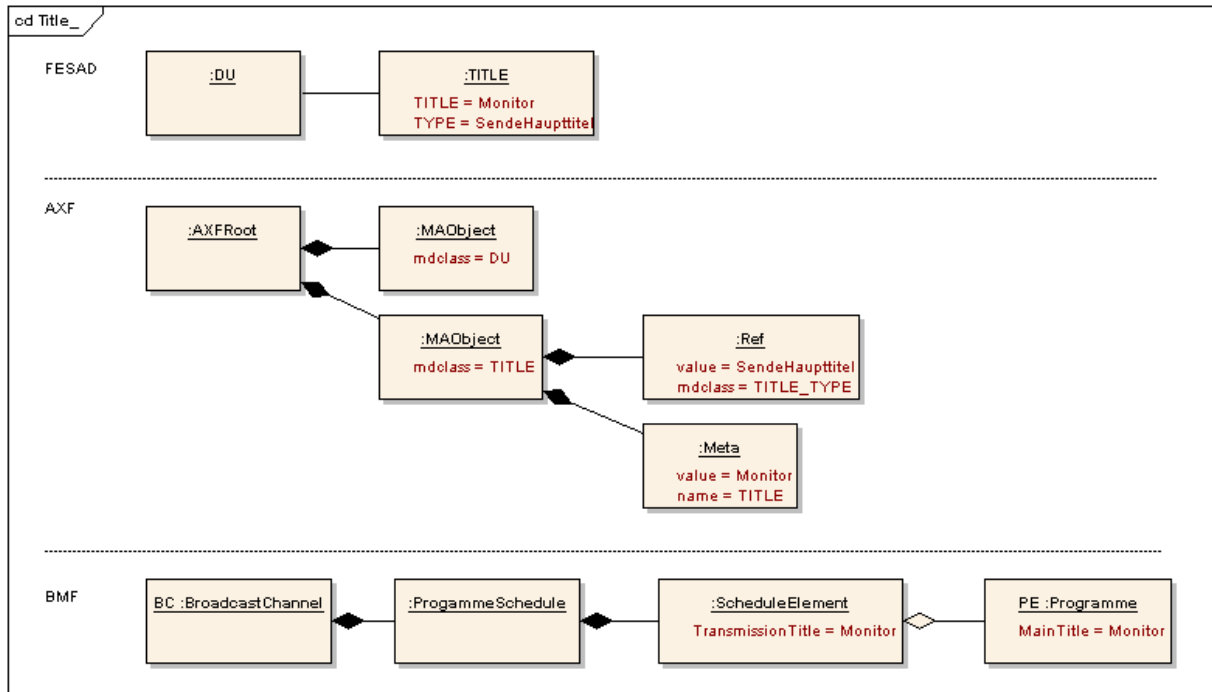


Abbildung 6.6: Abbildung eines *SendeHaupttitels*

6.2.1.4 Urheber, Produzent, Mitwirkende (APC)

Die Tabelle *APC*³ wird in BMF auf das Paket *Participant* abgebildet. Abb. 6.7 stellt die Modellierung einer Person und einer Organisation dar, die in der Produktion die Aufgaben der Regie bzw. Produktion übernehmen.

Das FESAD-Attribut *TYPE* der Tabelle *APC* spezifiziert die Tätigkeitsbeschreibung des Mitwirkenden (z.B. Regie) und damit die Einordnung dieser Tätigkeit in eine der Kategorien Urheber, Produzent, Mitwirkende und Sonstige.

Die Tätigkeitskategorien werden in BMF als Ableitungen der abstrakten Klasse *Participant* im Paket *Participant* realisiert. In den abgeleiteten Klassen werden die Tätigkeitsbeschreibungen der Teilnehmer über festgelegte Wertelisten (EnumerationLists) zugeordnet.

³APC: Author Production Contributor

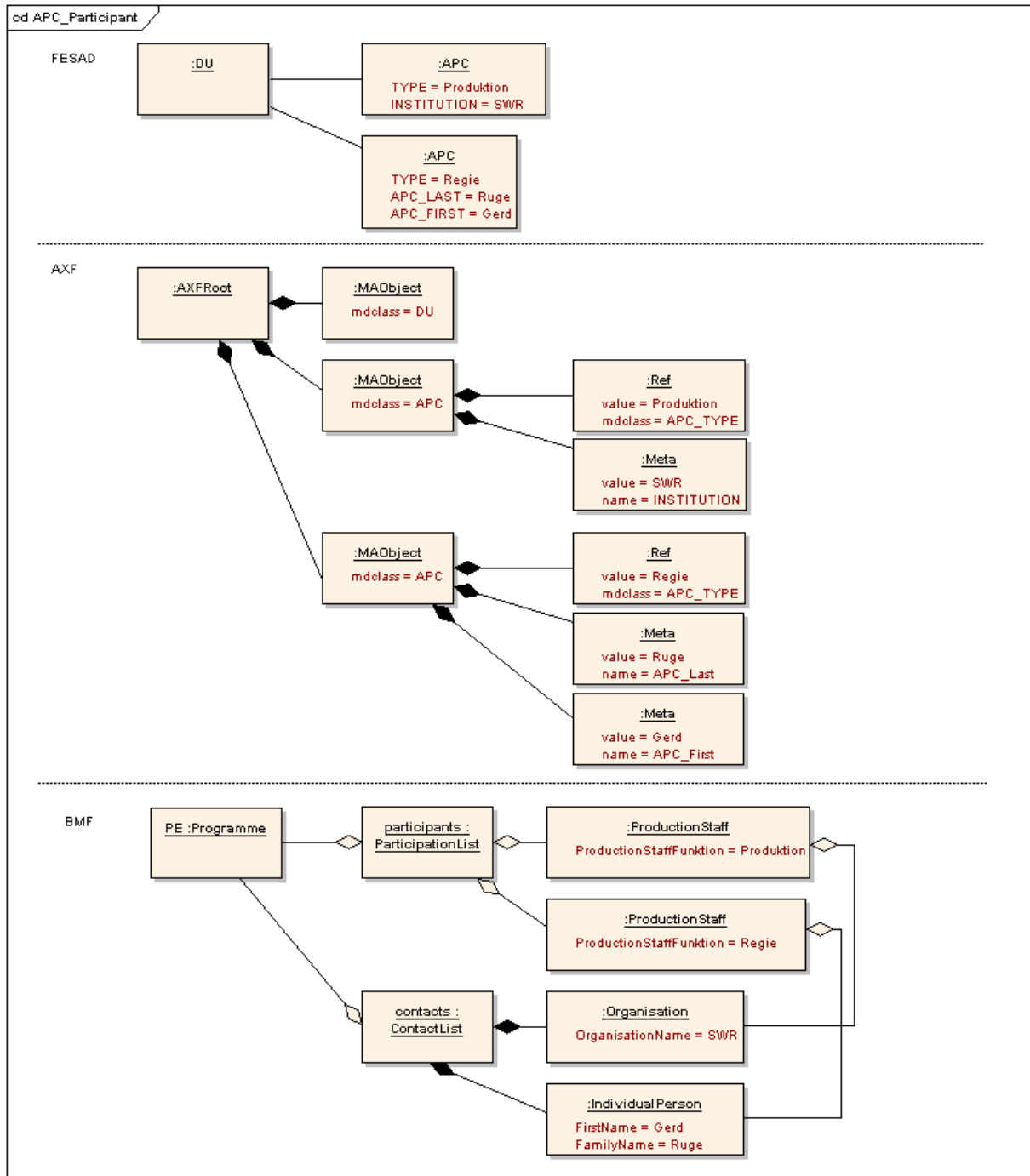


Abbildung 6.7: Abbildung von zwei Teilnehmern einer Produktion

Das je nach Tätigkeitskategorie unterschiedliche Mapping der Variable *TYPE* sieht wie folgt aus:

| FESAD | BMF | Bemerkungen |
|-------------|--|---|
| Urheber | participants:: Participants[] (Creator):: CreatorFunction | |
| Produktion | participants:: Participants[] (ProductionStaff):: ProductionStaffFunction | |
| Mitwirkende | participants:: Participants[] (Contributor):: ContributorFunction, participants:: Participants[] (Actor) | Schauspieler werden als eigenständige Klasse abgebildet |
| Sonstige | participants:: Participants[] (Others):: OthersFunction | |

Tabelle 6.7: Abbildung der Variable *TYPE*

Die Abbildung der Tabellenattribute der Klasse *APC* findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|-------------|---|---|
| TYPE | | siehe Tab. 6.24 |
| APC_FIRST | contacts:: LegalPersons[] (IndividualPerson):: FirstName | |
| APC_LAST | contacts:: LegalPersons[] (IndividualPerson):: FamilyName | |
| INSTITUTION | contacts:: LegalPersons[] (Organisation):: OrganisationName | |
| PSEUDONYM | contacts:: LegalPersons[] (IndividualPerson):: AlternateName | |
| CTLD_SUPP | - | Keine Abbildung in BMF |
| FREE_SUPP | - | |
| AWARDS | awards:: AwardList[]:: AwardName, awards:: AwardList[]:: FestivalDate, participants:: Participants[]:: Awards[] | Informationen werden nach Datum und Name des Festivals getrennt abgebildet. Weiterhin erhalten ausgezeichnete Teilnehmer eine Referenz auf den <i>Award</i> |
| REMARK | - | Keine Abbildung, da Anmerkungen an anderer Stelle im Modell möglich sind |

Tabelle 6.8: Abbildung der Tabelle *APC*

6.2.1.5 Inhaltsbeschreibung (CONTENT)

Die FESAD-Tabelle *CONTENT* kann Informationen zu Sachinhalt, Bildinhalt, O-Tönen, Musik, Fotos und Presstexten beinhalten. Auskunft über die Art der vorliegenden Informationen erteilt das Attribut *TYPE*.

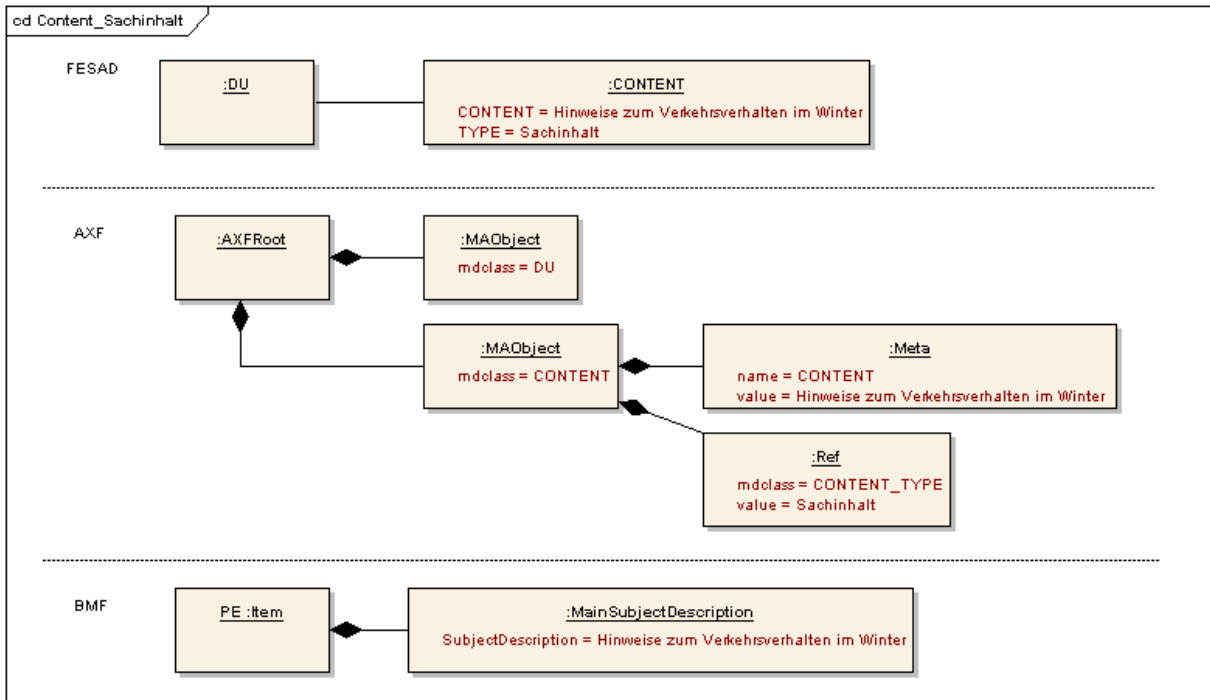
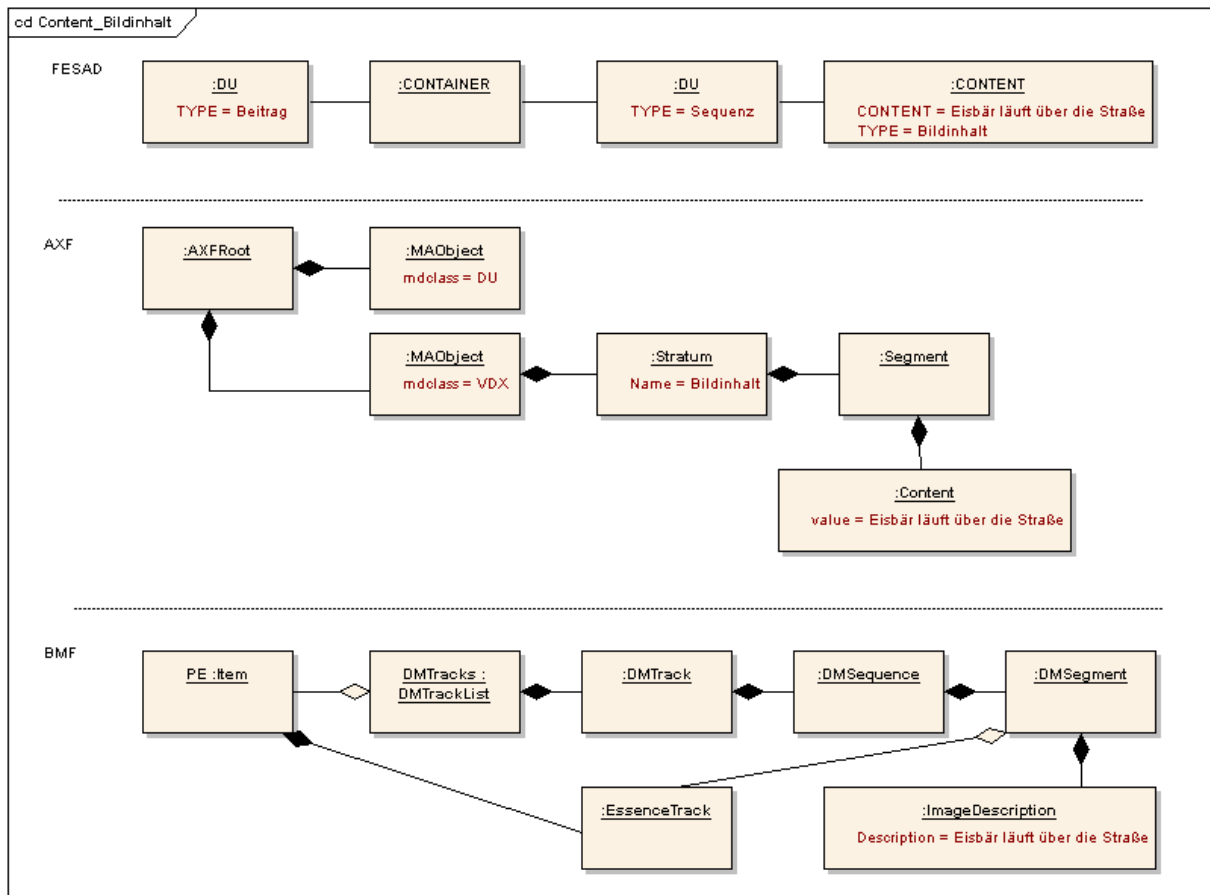


Abbildung 6.8: Abbildung von *CONTENT* im Falle von Sachinhalt

In BMF wurden diese Informationen aufgesplittet: Informationen zum Sachinhalt finden sich in der Klasse *MainSubjectDescription* direkt am Produktionselement, da Sachinhalt immer für die Zeitdauer eines gesamten Produktionselements gültig ist. Alle anderen Informationen finden sich im Paket *Sequence* (siehe Kap. 4.3.4.1), um auch Teilbereichen eines Produktionselements zugeordnet werden zu können. Abb. 6.8 stellt die Abbildung des Sachinhaltes, Abb. 6.9 die Abbildung eines Bildinhaltes dar. Alle weiteren o.g. Inhaltsbeschreibungstypen verhalten sich wie der Bildinhalt.

Abbildung 6.9: Abbildung von *CONTENT*

Die Abbildung der Tabellenattribute der Klasse *CONTENT* findet im Fall Sachinhalt wie folgt statt:

| FESAD | BMF | Bemerkungen |
|---------------|--|-------------------------------------|
| TYPE | | Nur Abbildung des Wertes Sachinhalt |
| CONTENT | PE:: MainSubjectDescriptions:: SubjectDescription | |
| FORMAT | PE:: MainSubjectDescriptions:: SubjectFormats | |
| CREATION_DATE | - | Für Austausch nicht relevant |
| CORR_DATE | - | |
| PROC | - | |

Tabelle 6.9: Abbildung der Tabelle *CONTENT* im Falle Sachinhalt

Die Abbildung der Tabellenattribute der Klasse *CONTENT* findet wie folgt statt (gilt nicht für den Informationstyp Sachinhalt):

| FESAD | BMF | Bemerkungen |
|---------------|--|---|
| TYPE | DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (ImageDescription), DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (InternationalSound), DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (Music), DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (Stills), DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (Presstext) | Die Abbildung der <i>TYPE</i> -Werte erfolgt mittels der von <i>ContentDescription</i> abgeleiteten Klassen, mit Ausnahme des Typs Sachinhalt |
| Content | DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (ContentDescription):: Description | |
| FORMAT | DMTracks:: DMTrackElements[]: DMSequence:: StructuralComponents[] (ContentDescription):: DescriptionFormats | |
| CREATION_DATE | - | Für Austausch nicht relevant |
| CORR_DATE | - | |
| PROC | - | |

Tabelle 6.10: Abbildung der Tabelle *CONTENT*

6.2.1.6 Kategorie (CATEGORY)

Die FESAD-Tabelle *CATEGORY* kann eine *DU* in den Kategorien Inhalt, Präsentationsform, Zielgruppe, Verwendung, Eignung und Sparte einteilen. In BMF sind diese Kategorien von der abstrakten Klasse *Category* abgeleitet. Die Werte, die eine Kategorie besitzen kann (z.B. bei Kategorie Zielgruppe: *Teenager*) werden in BMF mittels Enumerations festgelegt.

Abb. 6.10 stellt die Abbildung einer Kategorie Zielgruppe dar, die den Wert *Teenager* enthält.

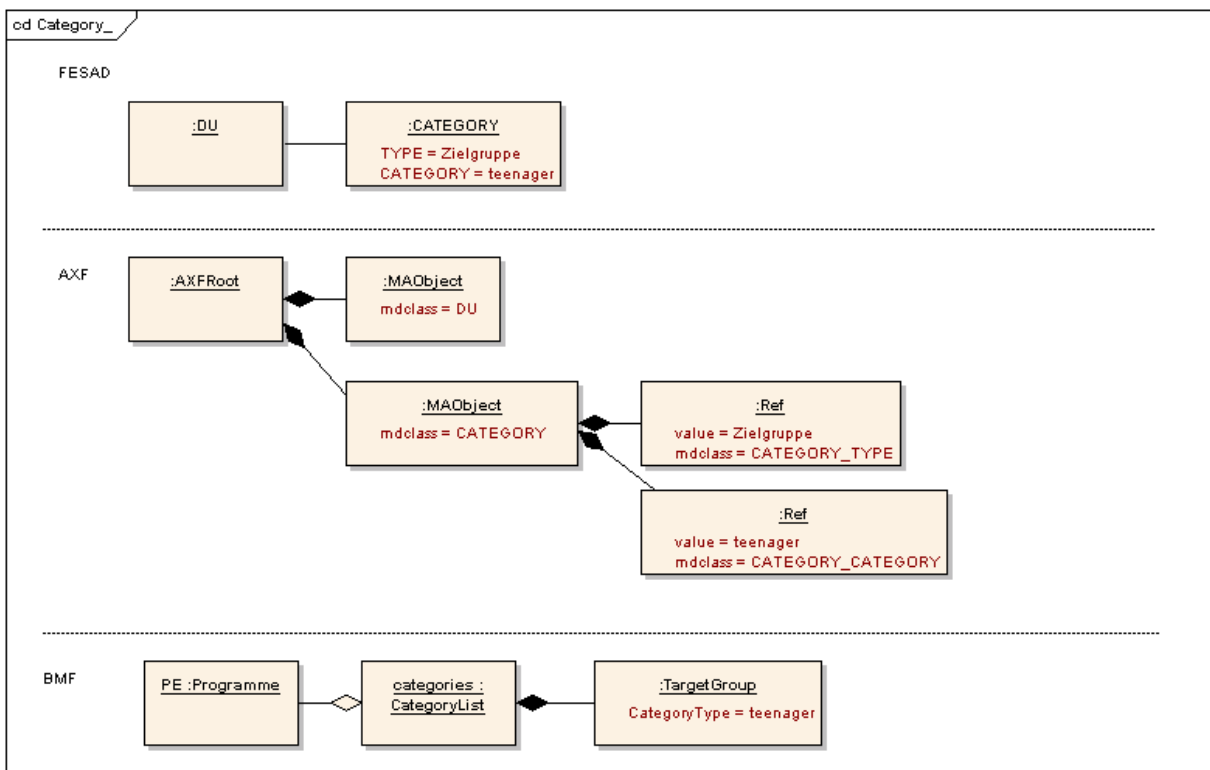


Abbildung 6.10: Abbildung der Tabelle *CATEGORY*

Das Mapping findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|----------|---|---|
| TYPE | categories:: CategoryElements[] (Content), categories:: CategoryElements[] (PresentationForm), categories:: CategoryElements[] (TargetGroup), categories:: CategoryElements[] (IntendedPurpose), categories:: CategoryElements[] (Section), categories:: CategoryElements[] (Suitability) | Die Abbildung der <i>TYPE</i> -Werte erfolgt mittels der von <i>Category</i> abgeleiteten Klassen |
| CATEGORY | categories:: CategoryElements[] (Content):: ContentType, categories:: CategoryElements[] (PresentationForm):: PresentationFormType, categories:: CategoryElements[] (TargetGroup):: TargetGroupType, categories:: CategoryElements[] (IntendedPurpose):: IntendedPurposeType, categories:: CategoryElements[] (Section):: SectionType, categories:: CategoryElements[] (Suitability):: EditorialSuitabilityLevel | |

Tabelle 6.11: Abbildung der Tabelle *CATEGORY*

6.2.1.7 Hauptdeskriptor (MD)

Die FESAD-Tabelle *MD* ermöglicht die Verschlagwortung einer *DU* in den Kategorien Allgemein, Themen, Orte, Personen und Zeit. In BMF wurden diese Kategorien als von *MainDescriptor* abgeleitete Klassen ausmodelliert.

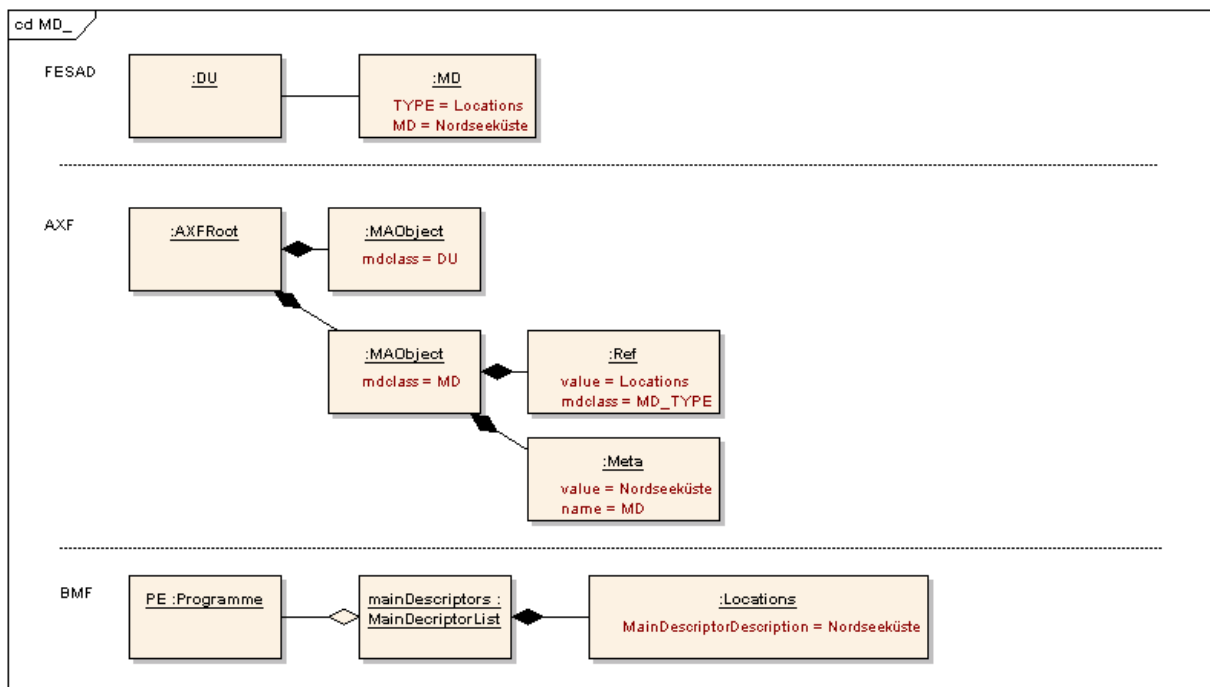
Abb. 6.11 stellt die Abbildung eines Hauptdeskriptors der Kategorie Orte dar.

Das Mapping findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|-------|---|---|
| TYPE | mainDescriptors:: MainDescriptorElements[] (General), mainDescriptors:: MainDescriptorElements[] (Subjects), mainDescriptors:: MainDescriptorElements[] (Locations), mainDescriptors:: MainDescriptorElements[] (Persons), mainDescriptors:: MainDescriptorElements[] (Age) | Die Abbildung der <i>TYPE</i> -Werte erfolgt mittels der von <i>MainDescriptor</i> abgeleiteten Klassen |

Fortsetzung

| FESAD | BMF | Bemerkungen |
|-------|---|-------------|
| MD | mainDescriptors:: MainDescriptorElements[]:: MainDescriptorDescription | |

Tabelle 6.12: Abbildung der Tabelle *MD*Abbildung 6.11: Abbildung der Tabelle *MD*

6.2.1.8 Anmerkung (REMARK)

Die FESAD-Tabelle *REMARK* ermöglicht es, Anmerkungen zu Elementen vom Typ *DU* einzupflegen. Das Mapping erfolgt abhängig vom jeweiligen in der Variable *TYPE* angegebenen Anmerkungstyp. In BMF finden sich die Anmerkungenfelder in den jeweiligen Klassen, auf die sie sich beziehen.

Tab. 6.13 gibt Auskunft über FESAD-Anmerkungstypen und deren Abbildung in BMF:

| FESAD | BMF | Bemerkungen |
|----------------------|---|---|
| Allgemeine | PE:: Remarks | |
| Dokumentarisch | PE:: PESTates (ArchiveAnnotation):: Remark | |
| Inhaltsbezogen | DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[] (ContentDescription):: Remark | |
| Kassation | DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[] (Restriction):: Remark | |
| Materialbezogen | contentInstances:: ContentInstanceList[]:: TechnicalAcceptance:: Remark | |
| Auszeichnung / Preis | - | Keine Abbildung, da durch Klasse <i>Award</i> hinreichend beschrieben |
| Begleitmaterial | additionalInformation:: AdditionalInformationElements[]:: Description | |
| Fertigmeldung | PE:: PESTates (PEProduced):: Remark | |
| Herkunft | production:: ProductionRemark | |
| KFG | production:: ProductionRemark | |
| MAZ-Karte | - | Keine Abbildung in BMF |
| PGG | - | |
| Projektgenehmigung | PE:: PESTates (PEApproved):: Remark | |
| Produktionsnummer | production:: ProductionRemark | |
| Rechte | rights:: Rights[]:: RightsRemarks | |
| Programmaustausch | order:: IntendedUse | |
| Weiterleitung | order:: IntendedUse | |
| WWW | internetRemark:: InternetPublicationElements[]:: InternetPublicationRemarks | |

Tabelle 6.13: Attribut *TYPE* der Tabelle *REMARK*

Abb. 6.12 stellt beispielhaft das Mapping einer Anmerkung vom Typ Allgemein dar.

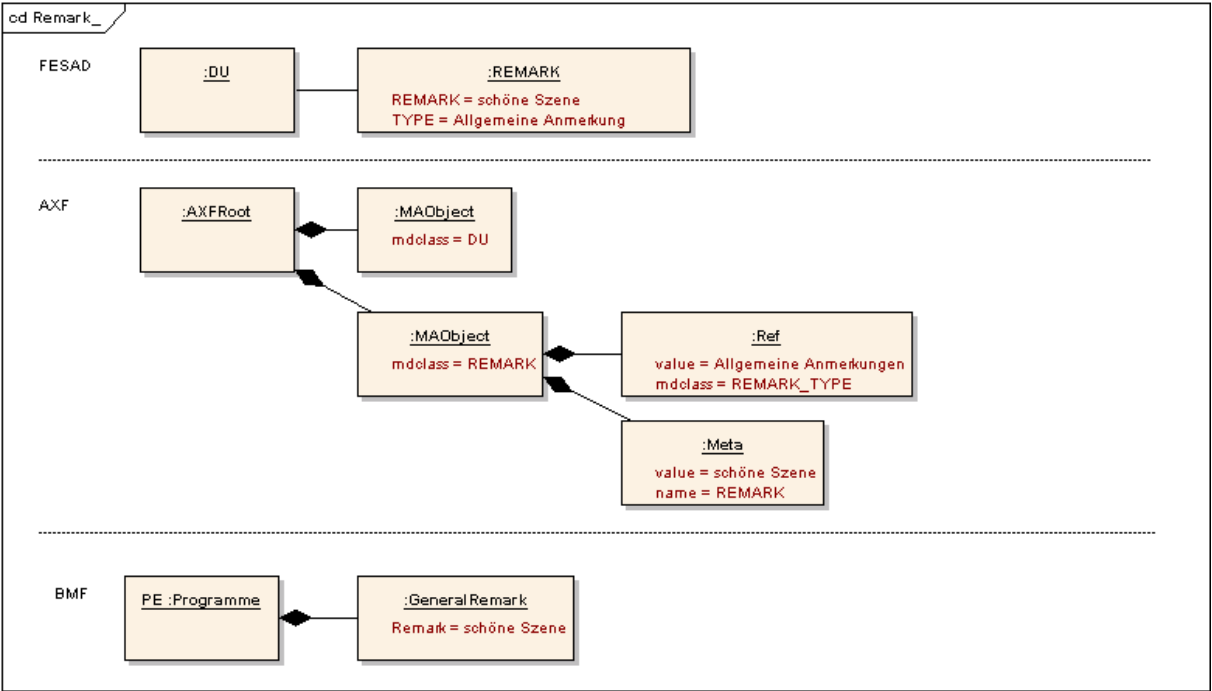


Abbildung 6.12: Abbildung der Tabelle *REMARK*

6.2.1.9 Lokale Zusatz-Info (INFO)

Die FESAD-Tabelle *INFO* ermöglicht die Anbindung von Zusatzinformationen, die binär in der Datenbank vorliegen. In BMF wurde hierzu das Paket *AdditionalInformation* entwickelt. In einem Objekt der Klasse *AdditionalInformationList* werden Elemente vom Typ der abstrakten Klasse *AdditionalInformation* abgespeichert. Von *AdditionalInformation* abgeleitete Klassen sind *BinaryInformation*, die Entsprechung zur FESAD-Tabelle *INFO*, und *ExternalInformation*, die Entsprechung zur FESAD-Tabelle *LINK* (siehe Kap. 6.2.1.10).

Das Mapping findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|-------|---|-------------|
| TYPE | <code>additionalInformation:: AdditionalInformationElements[]:: InformationType</code> | |
| MIME | <code>additionalInformation:: AdditionalInformationElements[] (BinaryInformation):: MimeType</code> | |

Fortsetzung

| | | |
|------------|--|-------------|
| FESAD | BMF | Bemerkungen |
| DESCR | additionalInformation:: AdditionalInformationElements[]: Description | |
| MULTIMEDIA | additionalInformation:: AdditionalInformationElements[] (BinaryInformation):: Information | |

Tabelle 6.14: Abbildung der Tabelle *INFO*

Abb. 6.13 stellt die Einbindung eines Presetextes dar.

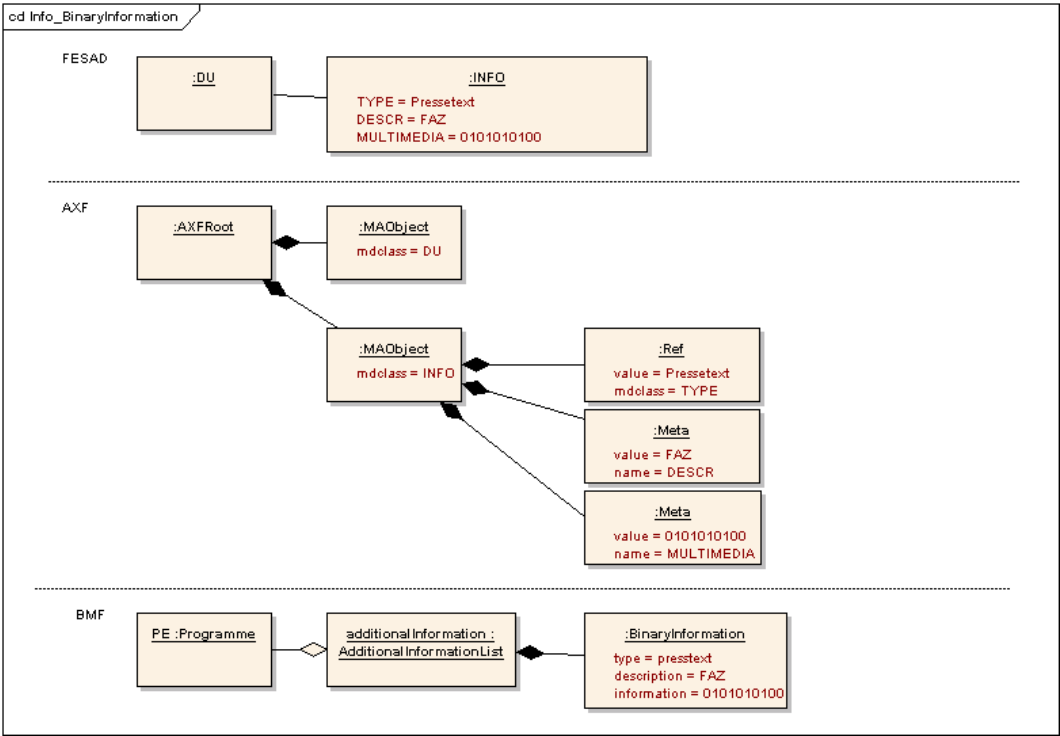


Abbildung 6.13: Abbildung der Tabelle *INFO*

6.2.1.10 Verknüpfung (LINK)

Die FESAD-Tabelle *LINK* ermöglicht es, auf extern vorliegendes Material zu verweisen, das als Zusatzinformation zu einer *DU* angesehen werden kann. In BMF wurde hierzu das Paket *AdditionalInformation* entwickelt. In einer Klasse *AdditionalInformationList* werden Elemente vom Typ der abstrakten Klasse *AdditionalInformation* abgespeichert. Von *AdditionalInformation* abgeleitete Klassen sind *ExternalInformation*, die Entsprechung zur FESAD-Tabelle *LINK*, und *BinaryInformation*, die Entsprechung zur FESAD-Tabelle *INFO* (siehe Kap. 6.2.1.9).

Abb. 6.14 stellt den Verweis auf eine Website eines Filmes dar.

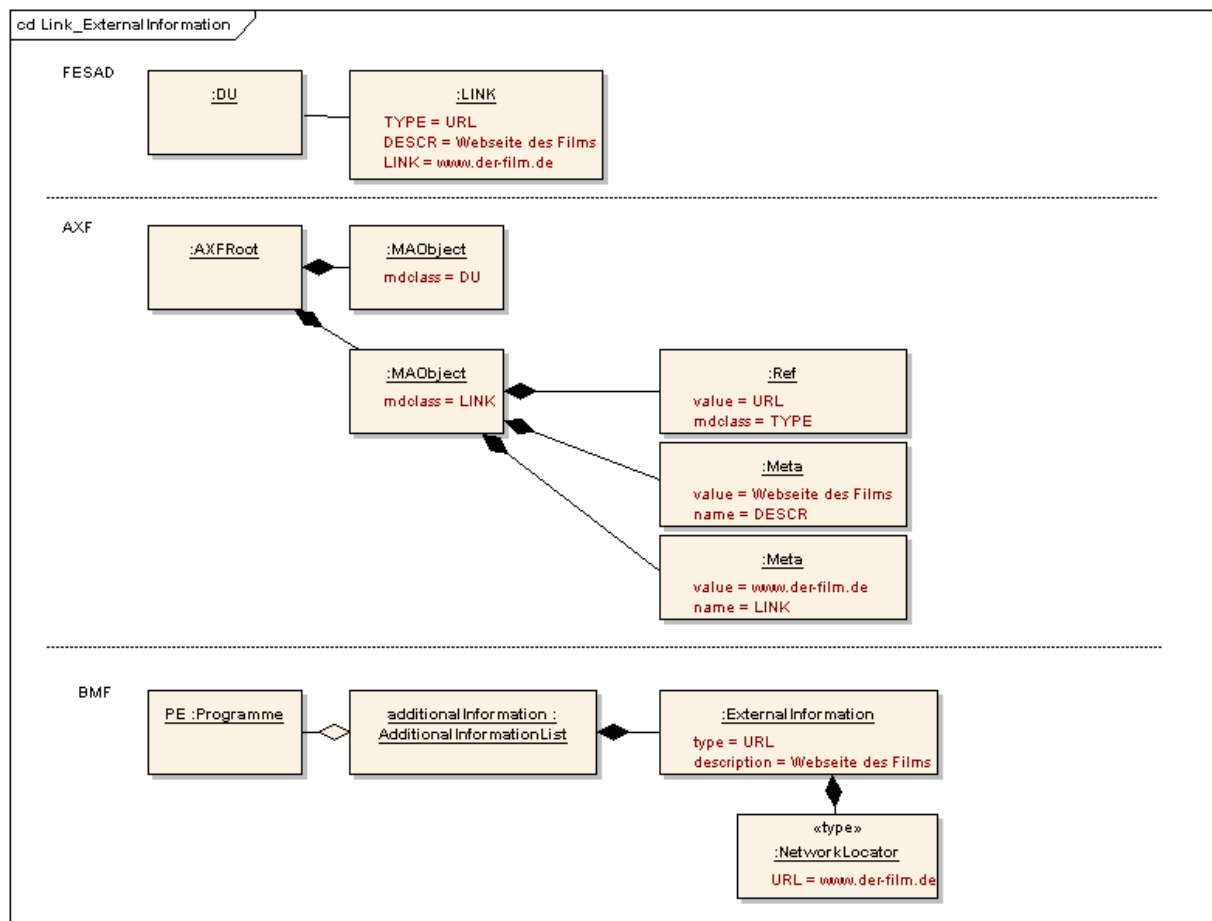


Abbildung 6.14: Abbildung der Tabelle *LINK*

Das Mapping findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|-------|---|---|
| TYPE | - | Angabe technisch nicht notwendig, daher keine Abbildung |
| LINK | additionalInformation:: AdditionalInformationElements[] (ExternalInformation):: Link | |
| DESCR | additionalInformation:: AdditionalInformationElements[]:: Description | |

Tabelle 6.15: Abbildung der Tabelle *LINK*

6.2.1.11 Rechte-Informationen (LICENCE)

Die FESAD-Tabelle *LICENCE* findet in den Rundfunkanstalten kaum oder keine Verwendung mehr. Beim BR werden Informationen zu Rechten, Honoraren und Lizenzen in einem externen Lizenzinformationssystem in Lizenzobjekten gespeichert. Der Verweis auf die entsprechenden, für das jeweilige Produktionselement relevanten, Lizenzobjekte wird in Kap. 6.2.2.5 behandelt.

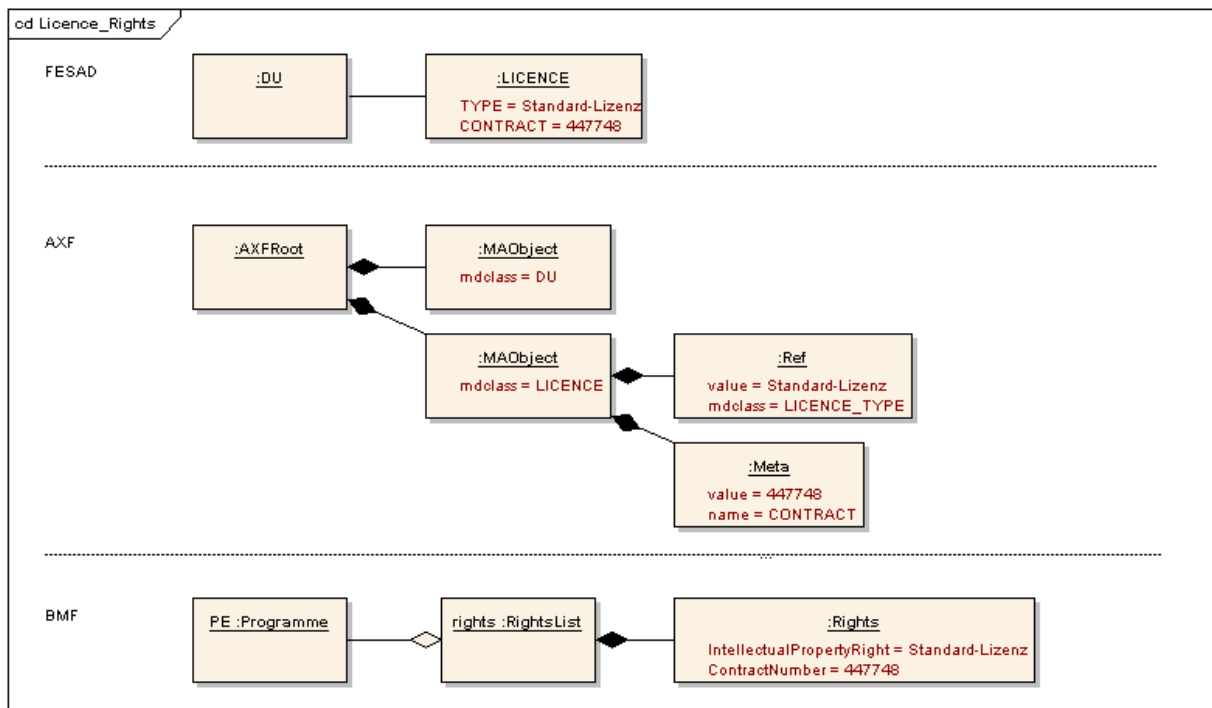
Um im Zweifelsfall jedoch ein Mapping zur FESAD-Tabelle zu ermöglichen, wurde in BMF die Klasse *Rights* eingeführt. Sie umfasst die Funktionalität der Tabelle *LICENCE* in FESAD und ermöglicht zusätzlich die Angabe eines Rechteinhabers über eine Weak-Reference (siehe Abb. 6.15).

Die Abbildung der Tabellenattribute der Klasse *LICENCE* findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|------------|---|-------------|
| TYPE | rights:: Rights[]:: IntellectualPropertyRight | |
| TEXT | rights:: Rights[]:: RightsDescription | |
| NO | rights:: Rights[]:: MaximumNumberOfUsage | |
| RANGE | rights:: Rights[]:: Range | |
| BEGIN | rights:: Rights[]:: RightsStart | |
| END | rights:: Rights[]:: RightsEnd | |
| OPTIONS | rights:: Rights[]:: RightsOption | |
| OPTION_EXP | rights:: Rights[]:: OptionExpireDate | |

Fortsetzung

| | | |
|----------|------------------------------------|-------------|
| FESAD | BMF | Bemerkungen |
| CONTRACT | rights:: Rights[]:: ContractNumber | |
| REMARK | rights:: Rights[]:: RightsRemarks | |

Tabelle 6.16: Abbildung *LICENCE* auf *Rights*Abbildung 6.15: Abbildung von *LICENCE* auf *Rights*

6.2.1.12 Beschränkung (RESTRICTION)

Die Abbildung der Tabelle *RESTRICTION* wird in Kapitel 6.2.2.1 erläutert und daher hier nicht nochmals aufgeführt.

6.2.1.13 Identifikationsnummer (ID-OF-DU)

Die FESAD-Tabelle *ID-OF-DU* ermöglicht die Angabe einer Produktions- und Archivnummer. Eine *DU* sollte zu jedem Zeitpunkt höchstens eine dieser Nummern besitzen. Eine Ausnahme dieser Regel stellt das Feld *PROD_NR* der FESAD-Tabelle *BC* dar, welches die Angabe einer neuen Produktionsnummer für eine Wiederholung ermöglicht. Archivaren zufolge werden in der Praxis der Wiederholung nur dann eine neue Produktionsnummer zugeordnet, wenn das Produktionselement bearbeitet wurde. In BMF würde in diesem Fall jedoch eine neue Instanz des Produktionselements angelegt werden, daher ist diese zweite Produktionsnummer irrelevant.

Beim BR wird die Tabelle *ID-OF-DU* nicht angewendet. Die BR-Produktionsnummer findet sich direkt im Attribut *ID* der Tabelle *DU* und die Archivnummer in der Tabelle *MEDIUM*, dessen Abbildung jedoch nicht Ziel dieser Diplomarbeit ist.

Abb. 6.16 stellt die Produktionsnummer im BR und die mögliche Verwendung der Tabelle *ID-OF-DU* dar.

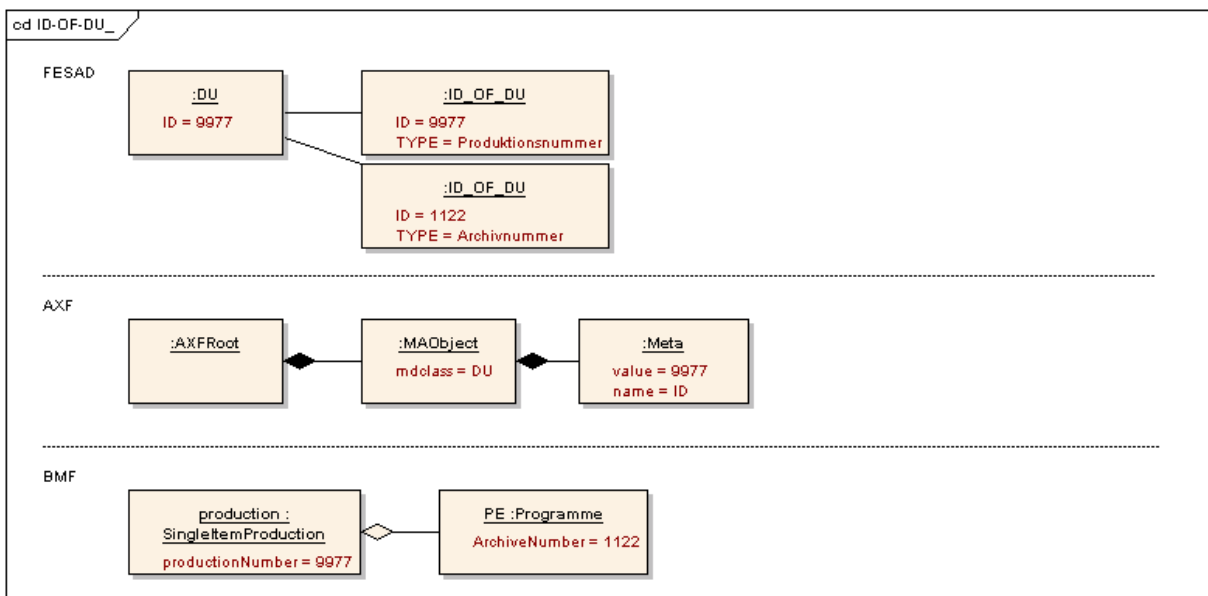


Abbildung 6.16: Abbildung von *ID-OF-DU*

Die Abbildung der Tabellenattribute der Klasse *ID-OF-DU* würde wie folgt stattfinden (für den BR nicht relevant):

| FESAD | BMF | Bemerkungen |
|-------|---|---|
| TYPE | production:: ProductionNumber, PE:: Archividentifizier | Je nach Wert der Variable <i>TYPE</i> befindet sich im Feld <i>ID</i> eine Produktions- oder Archivnummer |
| ID | | Wird zusammen mit Attribut <i>TYPE</i> abgebildet |

Tabelle 6.17: Abbildung der Tabelle *ID-OF-DU*

6.2.1.14 Ausstrahlungs- und Planungsdaten (BC)

Ausstrahlungs- und Planungsdaten finden sich in FESAD in der Tabelle *BC* und in BMF hauptsächlich in den Paketen *Programmplanung* und *Transmission*.

Das Attribut *TYPE* der FESAD-Klasse *BC* besitzt unterschiedliche Bedeutungsdimensionen, weshalb diese Informationen in BMF an unterschiedlichen Orten abgebildet werden. Laut BR-Informationen finden hausintern nur die Werte *Vorabausstrahlung*, *Wiederholung* und *Erstsendung* Verwendung.

Die Abbildung der *TYPE*-Werteliste findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|---|---|--|
| Anmeldung, Produktions-Planung | production:: ProductionState | |
| Aufzeichnungs-Beginn, Aufzeichnungs-Ende | productionPhases:: productionPhase[]:: PhaseType | Abbildung der Aufzeichnungsdaten in <i>StartDate</i> bzw. <i>EndDate</i> der Klasse <i>ProductionPhase</i> |
| Produktions-Ende | production:: ProductionStartDateandTime | |
| Produktions-Beginn | production:: ProductionEndDateandTime | |
| Sende-Planung, Vorraussichtliche Sendung | BC:: TransmissionSchedule:: ScheduleState | |
| Vorabausstrahlung, Erstsendung, Wiederholung | BC:: ScheduleAbstractElement:: TMEdition | |

Tabelle 6.18: Bedeutungsdimensionen des Attributes *TYPE* in *BC*

Die Abbildung der Tabellenattribute der Klasse *BC* findet wie folgt statt:

| FESAD | BMF | Bemerkungen |
|----------------|---|---|
| TYPE | | siehe Tab. 6.18 |
| STATION | BC:: ChannelName | |
| AREA | BC:: BroadcastArea | |
| AREA_SUPP | BC:: BroadcastArea-Supp | |
| TECHNOLOGY | BC:: BroadcastTechnique | |
| SIGNAL | BC:: BroadcastSignal | |
| SERVICE | BC:: BroadcastService | |
| DATE_BOR | BC:: ProgrammeScheduleArray[]:: BroadcastDay, BC:: ProgrammeScheduleArray[]:: ProgrammeWeek | |
| DATE_EOR | - | unscharfe Zeitangabe |
| TIME | BC:: ProgrammeScheduleArray[]:: ProgrammeScheduleElements[]:: TransmissionStartTime, BC:: ProgrammeScheduleArray[]:: TransmissionScheduleArray[]:: TransmissionScheduleElements[]:: TransmissionStartTime | Wert gilt für die Programm- und Sen- deplanung, daher doppelte Abbildung |
| EXACT_TIME | - | unscharfe Zeitangabe |
| DURATION | BC:: ProgrammeScheduleArray[]:: TransmissionScheduleArray[]:: TransmissionScheduleElements[]:: TransmissionScheduleDatas[]:: TransmissionScheduleData:: SourceDuration | |
| EXACT_DURATION | - | unscharfe Zeitangabe |
| SCHEMA | BC:: ProgrammScheduleArray[]:: ProgrammScheduleElements[] (ScheduleElement):: Schema | |
| REBC_TITLE | BC:: ProgrammeScheduleArray[]:: ProgrammeScheduleElements[]:: TransmissionTitle, BC:: ProgrammeScheduleArray[]:: TransmissionScheduleElements[]:: TransmissionTitle | Wert gilt für die Programm- und Sen- deplanung, daher doppelte Abbildung |

Fortsetzung

| FESAD | BMF | Bemerkungen |
|---------------|--|---|
| PART | grouping (Series):: SeriesProgrammeArray, grouping (Serial):: SerialProgrammeArray | Eine Aussage über die Reihen- / Folgennummer eines Produktionsele- mentes findet sich in BMF im Paket <i>Grouping</i> . Sie erfolgt durch die geord- nete Auflistung von Programmen in- nerhalb von Arrays bei von der Klasse <i>Grouping</i> abgeleiteten Klassen. |
| RATING | BC:: ProgrammeScheduleArray[]:: TransmissionScheduleArray[]:: TransmissionScheduleElements[] (TransmissionScheduleElement):: Rating | |
| MARKET_SHARE | BC:: ProgrammeScheduleArray[]:: TransmissionScheduleArray[]:: TransmissionScheduleElements[] (TransmissionScheduleElement):: MarketShare | |
| PROD_NR | production:: productionNumber | |
| COST_CENTER | BC:: ProgrammeScheduleArray[]:: ProgrammeScheduleElements[]:: costCenter, BC:: ProgrammeScheduleArray[]:: TransmissionScheduleElements[]:: costCenter | Wert gilt für die Programm- und Sen- deplanung |
| REMARK | BC:: ProgrammeScheduleArray[]:: ProgrammeScheduleElements[]:: TransmissionRemark, BC:: ProgrammeScheduleArray[]:: TransmissionScheduleElements[]:: TransmissionRemark | Wert gilt für die Programm- und Sen- deplanung |
| CREATION_DATE | - | Für Austausch nicht relevant |
| CORR_DATE | - | |

Tabelle 6.19: Abbildung BC auf *Transmission*

In FESAD ist die Angabe von unscharfen Zeit- und Datumsangaben möglich, d.h. eine Angabe von Bereichsgrenzen, wenn genaue Zeitwerte nicht bekannt sind. Dies geschieht mittels der Variablen *DATE_BOR*, *DATE_EOR*, *EXACT_TIME* und *EXACT_DURATION*. In BMF ist die Verbindlichkeit von Zeit- und Datumsangaben aus dem Status des Sende-

plans ersichtlich (Variable *ScheduleState* in der Klasse *TransmissionSchedule*). Ist diese auf *Scheduled* gesetzt, so ist die Datumsangabe als unscharfer Wert (Richtwert) zu verstehen, nicht so im Falle *OnAir* oder *AsRun*.

Abb. 6.17 stellt eine Wiederholungssendung dar. Anwendung finden die BMF-Pakete *Transmission* und *Programmplanung*.

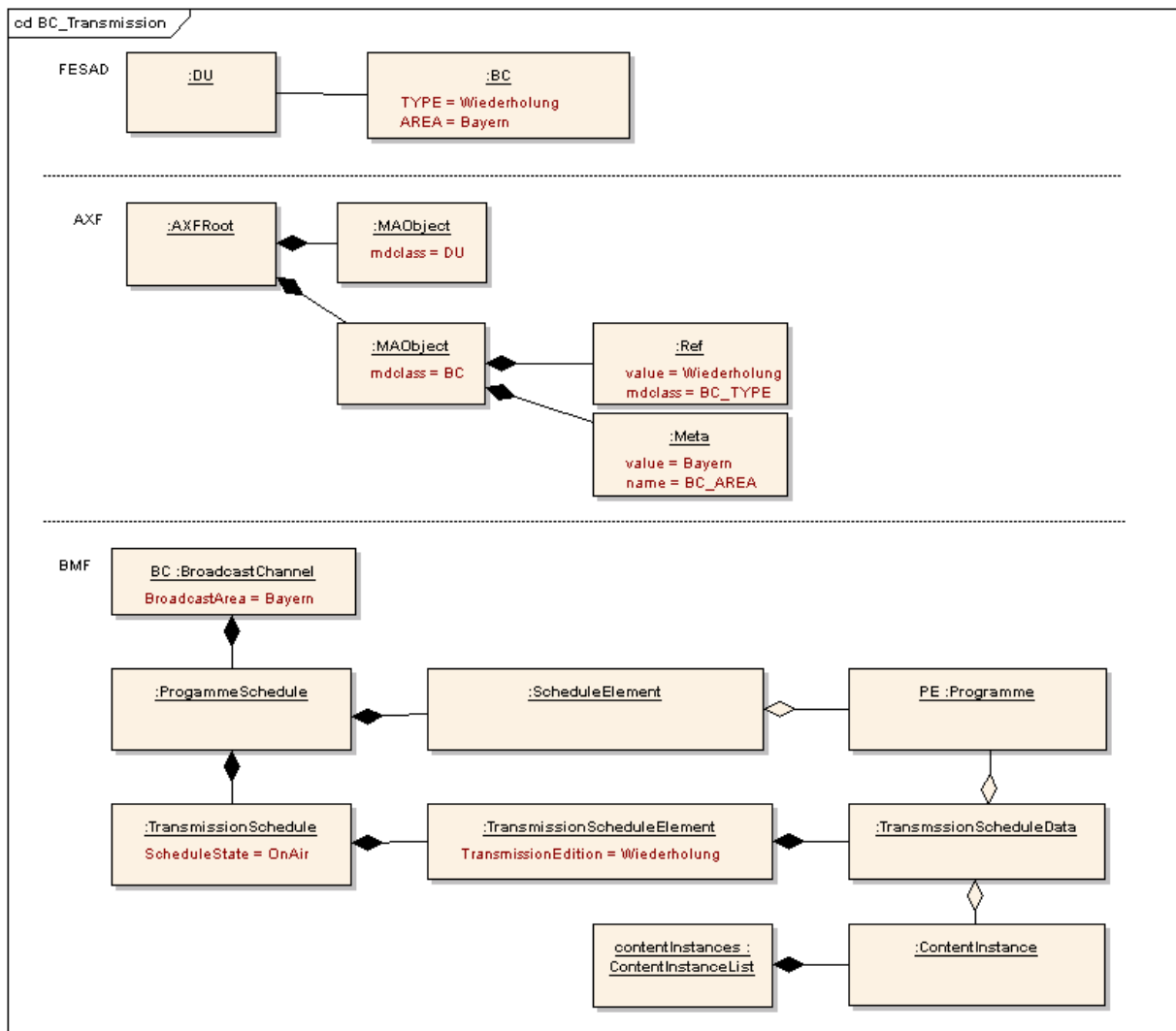


Abbildung 6.17: Abbildung von Ausstrahlungsdaten

6.2.1.15 Urheber, Produzent, Mitwirkende der Wiederholung (RAPC)

Die FESAD-Tabelle RAPC ermöglicht es, Informationen zu Mitwirkenden einer Wiederholung anzugeben.

In BMF gibt das Attribut *TMEdition* der Klasse *ScheduleAbstractElement* Auskunft über den Ausstrahlungsstatus einer Produktion, in diesem Fall also die Definition einer Wiederholung. Die Instanz des zu sendenden *Programme* enthält im Paket *Participant* Auskunft über UPM⁴-Daten. Durch Erstellung einer separaten Instanz für die Wiederholung eines *Programme* können so neue Angaben zu UPM-Daten gemacht werden.

⁴UPM: **U**rheber **P**roduktion **M**itwirkende

6.2.2 Abbildung Rechtesequenz auf Sequence

Die für den Rückgabewert der Funktion *getRightsequence* relevanten Rechtesequenzen des BR-Modells werden in BMF im Paket *Sequence* abgebildet. Als Rechtesequenzen sind dabei die AXF-Straten zu verstehen, die Rechteinformationen beinhalten (siehe Kap. 3.2.3.2). In FESAD lassen sich diese Rechteinformationen in der Tabelle *RESTRICTION* abbilden.

Wie in Kap. 3.2 und 4.3.4.1 vorgestellt, gibt es in AXF, FESAD und BMF die Möglichkeit, zeitlich definierte Abschnitte einer Essenz mit Metadaten zu beschreiben.

Alle in diesem Kapitel im Mapping angegebenen BMF-Attribute beziehen sich auf die Klasse *DMSegment*, die der Rückgabewert der Funktion *getRightsequence* ist.

Aufgrund der grundsätzlichen Unterschiede in der Darstellung von Sequenzen sei in der tabellarischen Darstellung dieses Kapitels auch AXF zur Verdeutlichung angefügt. AXF-Attributpfade folgen dem Schema (Knoten:: Knoten:: (...):: Attribut).

Abb. 6.18 stellt Rechtesequenzen einer Beitragsinstanz beispielhaft dar.

6.2.2.1 Beschränkung (RESTRICTION)

Die Abbildung der FESAD-Tabelle *RESTRICTION* findet wie folgt statt:

| FESAD | AXF | BMF |
|-----------------|---|--|
| TYPE | - | Metadata (BlockAccess), Metadata (ApplicationsRestriction), Metadata (Annulment) |
| IMPOSITION_DATE | - | Metadata (Restriction):: ImpositionDate |
| SUSPENSION_DATE | - | Metadata (Restriction):: SuspensionDate |
| TEXT | Content:: value in Stratum:: Bildinhalt-Rechtetext, Content:: value in Stratum:: O-Töne-Rechtetext | Metadata:: Remark |

Fortsetzung

| FESAD | AXF | BMF |
|-------|---|---|
| - | Content:: value in Stratum:: Bildinhalt-Rechtewert, Content:: value in Stratum:: O-Töne-Rechtewert | Metadata (Restriction):: RestrictionValue |

Tabelle 6.20: Abbildung der Tabelle *RESTRICTION*

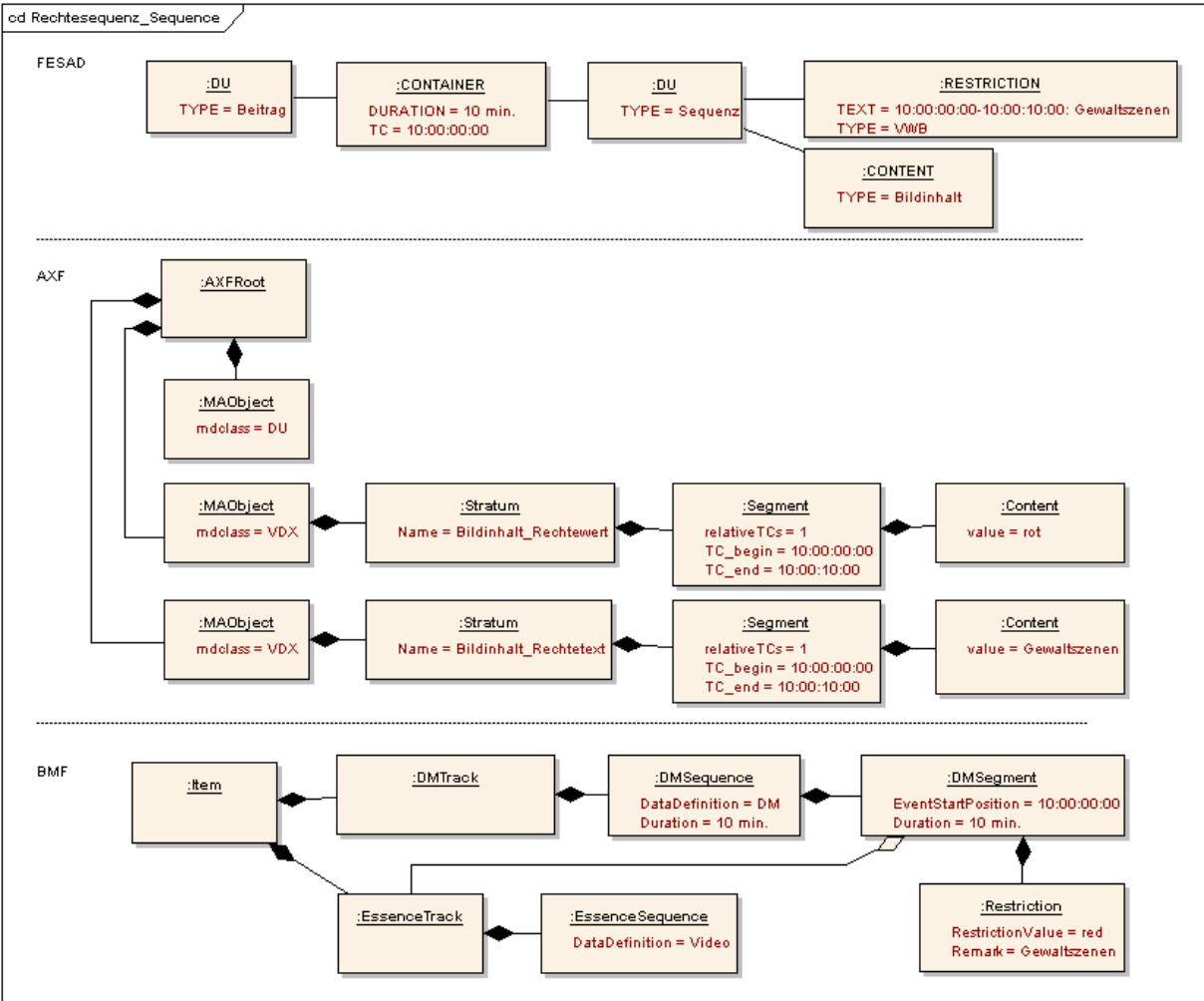


Abbildung 6.18: Abbildung von Rechtsequenzen

Das *TYPE*-Feld, welches in FESAD und BMF die Werte *Verwendungsbeschränkung*, *Sperre*, oder *Kassation* annehmen kann, gibt es in AXF nicht.

Auch die FESAD-Werte *IMPOSITION_DATE* und *SUSPENSION_DATE* werden in AXF nicht übertragen.

Die AXF-Variable *Content::value* in *Stratum::Bildinhalt-Rechtewert* oder in *Stratum::O-Töne-Rechtewert* stellt eine sog. „Rechte-Ampel“ dar, die die Werte rot, gelb, grün und grau annehmen kann und eine Aussage über die Verwendungsbeschränkung gibt. Diese Rechteampel existiert in FESAD nicht. Der BR-interne Rechteexport nach FESAD umfasst lediglich die rotmarkierten Bereiche.

In AXF erfolgt die Speicherung des Rechtetextes und Rechtewertes in unterschiedlichen Straten, je nachdem, ob es sich um Rechteinformationen zu O-Tönen oder Bildinhalt handelt.

6.2.2.2 Zeitliche Einbindung

Das Mapping der für die zeitliche Einbindung einer Sequenz erforderlichen Daten findet wie folgt statt:

| FESAD | AXF | BMF |
|--|--------------------|-------------------------------|
| Restriction:: Text und Container:: TC | Segment:: TC-Begin | EventStartPosition |
| Restriction:: Text und Container:: Duration+TC | Segment:: TC-End | EventStartPosition + Duration |

Tabelle 6.21: Abbildung zeitliche Parameter Rechtesequenzen

In FESAD werden *DU*-Sequenzen über die Klasse *CONTAINER* an ihre hierarchisch übergeordnete *DU* angebunden. Dieser Container definiert die Zeitdauer, für die die Rechteinformationen gelten können. Da eine Sequenz mehrere unterschiedliche Rechteinformationen beinhalten kann, ist es möglich, im Textfeld der Tabelle *RESTRICTION* den in *CONTAINER* angegebenen TC-Bereich nochmals aufzusplitten.

6.2.2.3 Gültigkeitsbereich (Bild oder Ton)

Rechtedaten können sich auf Bildinhalt oder O-Töne beziehen. In FESAD wird dies durch eine Tabelle *CONTENT* angegeben, die zusammen mit der Tabelle *RESTRICTION* einer Sequenz zugeordnet ist und im Feld *TYPE* die Werte *Bildinhalt* oder *O-Töne* annehmen kann.

In AXF erfolgt die Unterscheidung durch unterschiedliche Straten.

In BMF bezieht sich ein *DMSegment* und damit die Klasse *RESTRICTION* auf Elemente vom Typ *EssenceTrack*. Die Art (*DataDefinition*) dieser Klasse *EssenceTrack*, die in *EssenceSequence* angegeben wird grenzt den Gültigkeitsbereich ab.

6.2.2.4 Zuordnung der Sequenz zu übergeordnetem Element

Die Zuordnung einer Rechtesequenz zu ihrem hierarchisch übergeordneten Element erfolgt in FESAD mittels direkter Anbindung der Tabelle *RESTRICTION* an die *DU*. Sollen Rechteinformationen jedoch zeitlich segmentiert angegeben werden, so wird die Tabelle *RESTRICTION* an eine *DU* vom Typ Sequenz angebunden, die über einen *CONTAINER* auf die hierarchisch übergeordnete *DU* referenziert.

In AXF werden die hierarchisch übergeordnete *DU* und die Straten in einem gemeinsamen Objekt vom Typ *AXFRoot* übertragen und klären so die Zuordnung.

In BMF referenziert die Klasse *DMSegment* auf die betreffenden Elemente vom Typ *EssenceTrack*. Ein Objekt vom Typ *Restriction* wird durch *DMSegment* zeitlich spezifiziert und Objekten vom Typ *EssenceTrack* zugeordnet.

6.2.2.5 Referenzierung auf Lizenzobjekte

Beim BR werden Informationen zu Rechten, Honoraren und Lizenzen in einem externen Lizenzinformationssystem gespeichert. Dort existieren Lizenzobjekte für jede *DU* und für alle sich in einer *DU* befindlichen Rechtesequenzen.

Daher schien es sinnvoll, in BMF wie auch im BR-Datenmodell (bzw. AXF) einen entsprechenden Verweis auf die betreffenden Lizenzobjekte einzubauen. In BMF findet sich

dieser in der Tabelle *Restriction*, die wie bereits erwähnt Angaben zur Rechtesituation macht. Es ist so möglich, Lizenzobjekte für Rechtesequenzen anzugeben, oder aber für ein gesamtes Produktionselement durch Setzen der Variable *Duration* in *DMSegment* auf „null“.

In AXF besteht zum jetzigen Zeitpunkt noch keine Variable für diesen Verweis. Eine Möglichkeit wäre, wie in Abb. 6.19 dargestellt, die Einführung einer neuen Variable *LicenceObjectKey* im Element *Segment*.

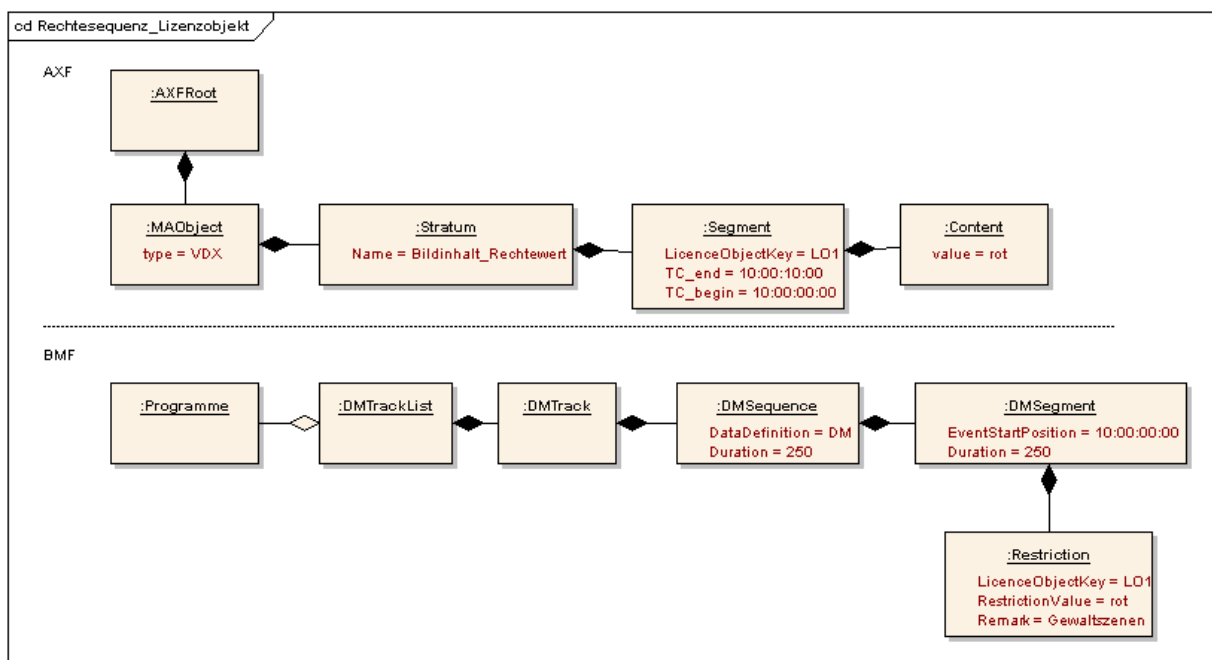


Abbildung 6.19: Referenzierung auf ein Lizenzobjekt

6.2.3 Abbildung Order auf FullOrder / StandardOrder

Die für den Rückgabewert der Funktionen *getFullOrder* und *getStandardOrder* relevanten Objekte vom Typ *Order* (siehe Kap. 3.2.3.3) werden in BMF auf Objekte vom Typ *FullOrder* und *StandardOrder* abgebildet.

Alle in diesem Kapitel im Mapping angegebenen BMF-Attribute beziehen sich daher auf die Klassen *FullOrder* und *StandardOrder*.

Da es in FESAD und AXF keine Elemente vom Typ *Order* gibt, werden diese Datenmodelle im vorliegenden Kapitel nicht abgebildet.

Abb. 6.20 stellt die Klassen *FullOrder* und *StandardOrder* dar. Sie unterscheiden sich lediglich in der Einbindung von *DU*-Elementen. In der Klasse *FullOrder* werden zur Repräsentierung der *DU* Objekte vom Typ *PEFull*, in *StandardOrder* Objekte vom Typ *PEStandard* eingebunden.

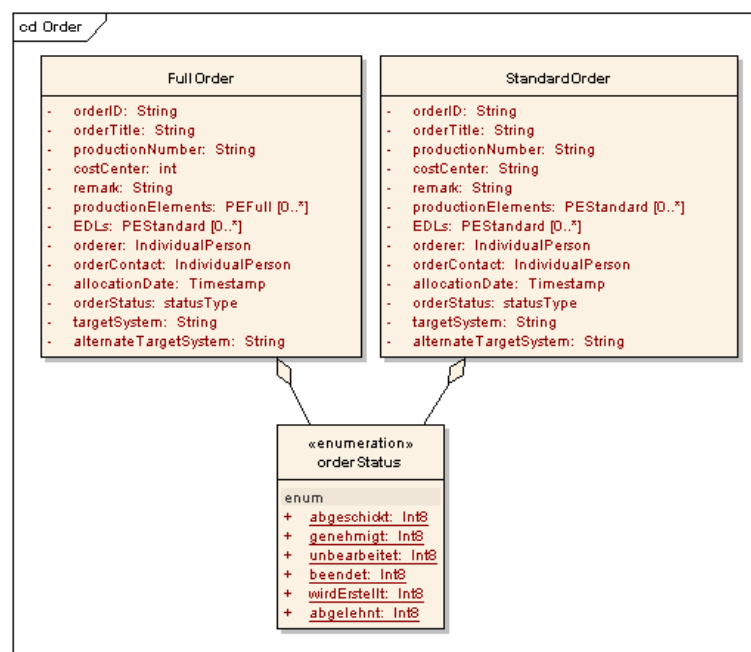


Abbildung 6.20: Die Klassen *FullOrder* und *StandardOrder*

Das Mapping der BR-Tabelle *Order* findet wie folgt statt (sog. Kopfdaten):

| BR-Datenmodell | BMF | Bemerkungen |
|-------------------------------|---|--|
| Auftragsnummer | orderId | |
| Titel der Produktion | orderTitle | |
| Produktionsnummer | productionNumber | |
| Kostenstelle | costCenter | |
| Besteller | orderer:: FamilyName | |
| Bestellredaktion | orderer:: Organisations[]:: Departments[]:: DepartmentName | |
| Bestellkontakt | orderContact:: FamilyName, orderContact:: FirstName | |
| Mailadresse des Bestellers | orderer:: Addresses:: EmailAddress | |
| Bereitstellungstermin | allocationDate | |
| Bestell-Status | orderStatus | |
| Zielsystem | targetSystem | |
| Alternatives Zielsystem | alternateTargetSystem | |
| Anmerkungen (OMA) | Remark | |
| RD-Event-ID | - | Befindet sich in BMF in Entwick- lung |
| TS-Event-ID | productionElements[]:: PE:: BC:: ProgrammScheduleArray[]:: ProgrammeScheduleElement[]:: InstanceUID | Diese Angabe bezieht sich auf alle in einer <i>Order</i> vorhandenen Elemente, die später auf einem gemeinsamen Sendepplatz ausge- strahlt werden sollen |
| Formatkennung | - | Wird beim BR derzeit nicht ge- nutzt |

Tabelle 6.22: Abbildung der BR-Tabelle *Order* (Kopfdaten)

Eine *Order* kann Objekte vom Typ *DU* und EDL umfassen. Während Elemente vom Typ *DU* wie besprochen in den Objekten *PEFull* und *PEStandard* abgebildet werden, die in den Klassen *FullOrder* und *StandardOrder* im Array *productionElements[]* eingebunden sind, wurde die Abbildung von EDL-Objekten noch nicht explizit angesprochen.

Im Unterschied zur hierarchischen Zusammensetzung von Elementen vom Typ *DU* imple-

mentieren EDL-Objekte den Track-Mechanismus, der es erlaubt, einem Produktionselement mehrere Tracks (Signale) zuzuweisen. Dieser Mechanismus wird bei BMF im Kap. 4.3.1.1 erläutert, in AXF und FESAD kann er bisher nicht angewendet werden. Die abzubildende EDL liegt beim BR als Datensatz im CMS und als XML-Datei mit eigenem Schema vor, nicht also im AXF-Format.

In BMF wurde zur Abbildung einer EDL die Klasse *PEStandard* wiederverwendet, die ein übergeordnetes Produktionselement (*PE*) besitzt, welches alle verwendeten Tracks beinhaltet. Im XML-Knoten *Edit* referenzierte Objekte werden im Array *SubPEList[]* der Klasse *PEStandard* eingebunden. Das übergeordnete Produktionselement *PE* enthält zur eindeutigen Kennzeichnung als EDL in der Variable *PEStates* den Status *PERoughEditList*.

Zusätzlich zur Abbildung des EDL-Datensatzes auf die BMF-Struktur wird die EDL in Form einer XML-Datei in das übergeordnete Produktionselement *PE* der Klasse *PEStandard* eingebunden. Der genaue Ort lautet *EDLs:: PE:: additionalInformation:: AdditionalInformationElements[] (ExternalInformation):: Link*.

Je nach Anwendungsfall kann die EDL so entweder als extern vorliegende Datei oder als auf BMF abgebildete Datenstruktur weiter verwendet werden.

Abb. 6.21 stellt eine EDL (*Editlist*) mit repräsentiertem Videosignal (*Track*) und einem Schnitt (*Edit*) in der BR-XML-Datei und dessen Abbildung auf BMF dar.

Das Mapping einer EDL (*Editlist*) findet wie folgt statt:

| BR-Datenmodell | BMF | Bemerkungen |
|-------------------|---|---|
| ID | EDLs[]:: PE:: PE:: InternalProductionIdentifier | |
| Framerate | EDLs[]:: PE:: PE:: Tracks[]:: Editrate | Angabe der Framerate für jedes Element vom Typ <i>Track</i> einzeln möglich |
| Title | EDLs[]:: PE:: PE:: MainTitle | |
| Owner | - | Für den Austausch nicht von Bedeutung |
| Creation_Date | - | |
| Modification_Date | - | |

Tabelle 6.23: Abbildung einer EDL

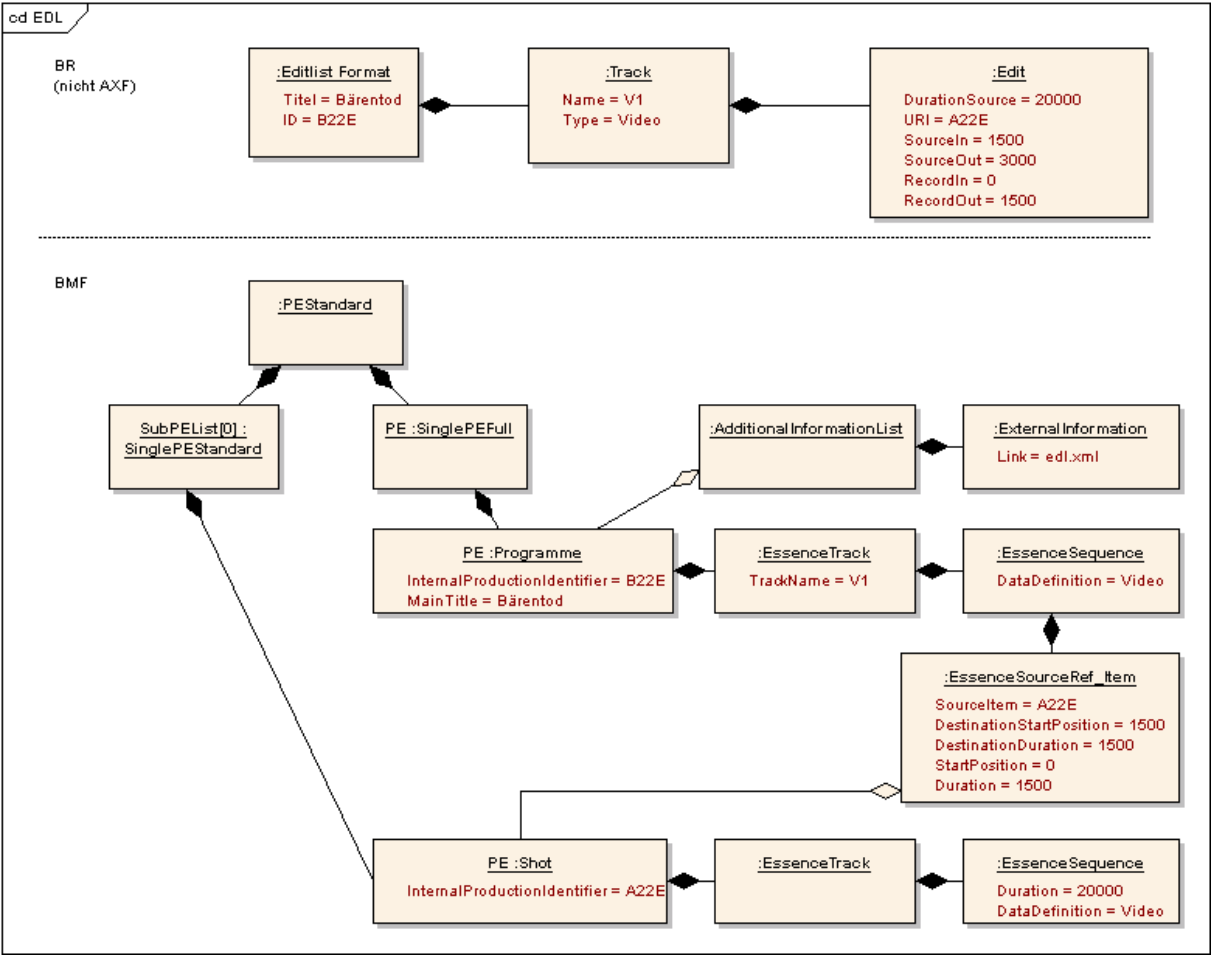


Abbildung 6.21: Abbildung einer *EDL*

Das Mapping eines Signal-Knotens (*Track*) einer EDL findet wie folgt statt:

| BR-Datenmodell | BMF | Bemerkungen |
|----------------|---|---|
| Name | EDLs[]:: PE:: PE:: Tracks[]:: TrackName | |
| Type | EDLs[]:: PE:: PE:: Tracks[]:: Sequence:: DataDefinition | |
| Label | EDLs[]:: PE:: PE:: Tracks[]:: TrackName | Bedeutung konnte nicht geklärt werden, daher vorerst Abbildung auf <i>TrackName</i> |

Fortsetzung

| BR-Datenmodell | BMF | Bemerkungen |
|-------------------|---|---|
| Comment | EDLs[]:: PE:: DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[]:: Metadata (ImageDescription):: Description, EDLs[]:: PE:: DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[]:: Metadata (InternationalSound):: Description | Abbildung des Kommentars auf das Attribut <i>Description</i> in den Klassen <i>ImageDescription</i> und <i>InternationalSound</i> , je nachdem, um welchen Signaltyp (<i>Track</i>) es sich handelt. Da diese deskriptiven Metadaten für die Gesamtheit eines Signals gelten, referenziert <i>DMSegment</i> auf dieses Signal (<i>Track</i>) und setzt die <i>Duration</i> auf „null“. |
| Active | | Bedeutung konnte nicht geklärt werden, daher vorerst keine Abbildung in BMF |
| DisplayInTimeline | | |
| DisplayInCutlist | | |

Tabelle 6.24: Abbildung eines Signals einer EDL

Das Mapping eines Schnittes (*Edit*) eines Signals findet wie folgt statt:

| BR-Datenmodell | BMF | Bemerkungen |
|----------------|---|--|
| URI | EDLs[]:: SubPEList:: contentInstances:: ContentInstanceList:: contentStorage:: packages:: descriptor:: Locators (NetworkLocator):: URL | Die Variable <i>URL</i> der Klasse <i>NetworkLocator</i> referenziert auf die Essenz |
| Title | EDLs[]:: SubPEList[]:: PE:: MainTitle | |
| Timebase | EDLs[]:: SubPEList[]:: PE:: Tracks[]:: Origin | |
| SourceIn | EDLs[]:: PE:: PE:: Tracks[]:: Sequence:: structuralComponents[]:: Startposition | |
| SourceOut | EDLs[]:: PE:: PE:: Tracks[]:: Sequence:: structuralComponents[]:: Startposition + Duration | |
| RecordIn | EDLs[]:: PE:: PE:: Tracks[]:: Sequence:: structuralComponents[]:: DestinationStartPosition | |

Fortsetzung

| BR-Datenmodell | BMF | Bemerkungen |
|----------------|--|--|
| RecordOut | EDLs[]:: PE:: PE:: Tracks[]:: Sequence:: structuralComponents[]:: DestinationStartPosition + DestinationDuration | |
| DurationSource | EDLs[]:: SubPEList[]:: PE:: Tracks[]:: Sequence:: Duration | Der Wert bezieht sich auf das in <i>EDLs[]:: PE:: PE:: Tracks[]:: Sequence:: structuralComponents[]</i> einbezogene Produktionselement |
| Streamtype | EDLs[]:: PE:: contentInstances:: contentInstanceList[]:: contentStorage:: packages[]:: descriptor (GenericPictureEssenceDescriptor):: PictureEssenceCoding | |
| Comment | EDLs[]:: PE:: DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[]:: Metadata (ImageDescription):: Description, EDLs[]:: PE:: DMTracks:: DMTrackElements[]:: DMSequence:: StructuralComponents[]:: Metadata (InternationalSound):: Description | Abbildung des Kommentars auf das Attribut <i>Description</i> in den Klassen <i>ImageDescription</i> und <i>InternationalSound</i> , je nachdem, um welchen Signaltyp (<i>Track</i>) es sich handelt. |
| TapelD | - | Bisher in BMF nicht vorhanden |
| ArchiveID | EDLs[]:: SubPEList[]:: PE:: ArchiveIdentifier | |
| ssmbaid | - | Bedeutung konnte nicht geklärt werden, daher vorerst keine Abbildung in BMF |
| ipvoffset | - | |
| msoffset | - | |
| beroffset | - | |

Tabelle 6.25: Abbildung eines Schnittes einer EDL

Kapitel 7

Zusammenfassung und Ausblick

Thema dieser Diplomarbeit war der Entwurf einer Integrationsschnittstelle für Content-Management-Systeme basierend auf dem Datenmodell BMF.

Im konkreten Fall handelte es sich um das CMS des Bayerischen Rundfunks, dessen Metadatenmodell auf dem der Fernseharchivdatenbank FESAD basiert und dessen Daten über das proprietäre AXF-Format ausgetauscht werden können.

In der zu entwerfenden Schnittstelle sollten CMS-Daten ausgegeben und in die Struktur des Datenmodells BMF transformiert werden. Dazu mußte zunächst der tatsächliche Informationsgehalt des BR-Datenmodells festgestellt werden, da dieses nicht veröffentlicht ist. Durch Analyse einer AXF-Transformationsdatei zeigte sich, daß das BR-Datenmodell zwar große Übereinstimmung zu FESAD besitzt, jedoch auch einige wesentliche Abweichungen.

Diese Erkenntnis hatte zur Folge, daß BMF, FESAD und das BR-Datenmodell, repräsentiert durch AXF, im weiteren Verlauf der Arbeit wenn erforderlich parallel betrachtet wurden, um die Schnittstelle nicht nur speziell für den BR, sondern auch für allgemeine FESAD-Anwendungen zu spezifizieren.

Da der analysierte Informationsumfang anfangs nicht gänzlich auf BMF abgebildet werden konnte, wurden im Rahmen dieser Diplomarbeit einige BMF-Pakete neu modelliert und Bestehende weiterentwickelt.

Der BR stellte daraufhin Anforderungen an die Funktionalität der Schnittstelle, die es zu erfüllen galt. Der letztendliche Entwurf der Schnittstelle umfasst Zugriffsmethoden und die für die Rückgabewerte dieser Methoden relevanten Datentransformationen.

Abschließend kann festgehalten werden, daß das Ziel dieser Arbeit, eine Schnittstelle gemäß den vom BR gestellten Anforderungen zu entwickeln, erreicht wurde. Im nächsten Schritt des Projektes kann das Dargestellte implementiert, also praktisch umgesetzt, werden. Dies gilt sowohl für den BR als auch für andere, auf FESAD-basierende, Systeme.

Für BMF ist diese Arbeit ein erster Praxistest und der Kompatibilitätsnachweis auf das weit verbreitete FESAD-Modell der ARD. Es scheint erstrebenswert, BMF nun schrittweise in den öffentlich-rechtlichen Rundfunkanstalten einzusetzen und so Herstellern einen Anreiz zu geben, das Modell zu implementieren. Eine SMPTE-Standardisierung für deskriptive Metadaten des Material Exchange Formats (MXF) erscheint ebenfalls sinnvoll. Ziel sollte es auch im Sinne der öffentlich-rechtlichen Rundfunkanstalten sein, BMF als herstellerunabhängigen, soliden Standard zu etablieren, um problemlosen Austausch von Metadaten zu ermöglichen.

Literaturverzeichnis

- [1] DE JONG, ANNEMIEKE: *Metadaten in der audiovisuellen Produktionsumgebung*. FKT, die Fachzeitschrift für Fernsehen, Film und elektronische Medien, Nr. 8-9 / 2001:485–495, 2001.
- [2] DEUTSCHES RUNDFUNK ARCHIV, Betramstr.8, Frankfurt a. M.: *Regelwerk Fernsehen*, 1998.
- [3] EBERHARDT, ANDREAS und STEFAN FISCHER: *Web Services, Grundlagen und praktische Umsetzung mit J2EE und .NET*. Hanser, 2003.
- [4] EBNER, ANDREAS: *Austausch von Metadaten - BMF*. Technischer Bericht B193/2005, Institut für Rundfunktechnik, München, 2005.
- [5] EBNER, ANDREAS und HUBERT HUBBES: *BMF, Beschreibung des Klassenmodells, Draft*. Technischer Bericht, Institut für Rundfunktechnik, München, 2006.
- [6] ENGLMAIER, WOLFGANG: *Data-Mining im BR-SZFM*, Mai 2006. Powerpoint-Präsentation, Vertraulich.
- [7] RÖSCH, PETER: *Media Archive Version 3.0, Attribute eXchange Format Specification, internal/confidential*. Technischer Bericht, Blue Order AG, Kaiserslautern, Januar 2006.
- [8] SÜDWESTRUNDFUNK UND HESSISCHER RUNDFUNK: *Entity-Relationship-Diagramm FESAD*, 2006. Verfügbar im Corporate Network der ARD unter <http://fesad.ivz.cn.ard.de/documentation/erd/fesad/index.html>.
- [9] ZIMMERMANN, RICO: *Transformation des Datenmodells BMF in das implizite Datenmodell von MXF*. Diplomarbeit, Technische Universität Illmenau, Juni 2006.

Abbildungsverzeichnis

| | | |
|------|--|----|
| 2.1 | Metadaten in einem beispielhaften Fernsehproduktionsworkflow | 4 |
| 2.2 | Aggregation und Komposition | 6 |
| 2.3 | Referenzierungsarten | 6 |
| 3.1 | Das digitale Archiv im BR (siehe [6]) | 9 |
| 3.2 | Beispielhafter Auszug der FESAD-Datenstruktur | 11 |
| 3.3 | Hierarchieebenen in FESAD | 12 |
| 3.4 | Beispielhafter Auszug der AXF-Datenstruktur | 13 |
| 3.5 | Hierarchieabbildungen in AXF | 15 |
| 3.6 | Straten in AXF | 16 |
| 3.7 | Schnittstellen im CMS des BR | 17 |
| 4.1 | Beispielhafter Prozessablauf für eine Fernsehproduktion [4] | 19 |
| 4.2 | Hierarchische Struktur zur Beschreibung von Produktionselementen | 21 |
| 4.3 | Vererbungshierarchie der Produktionselemente | 24 |
| 4.4 | Struktureller Aufbau einer Sendung mit einem Beitrag | 25 |
| 4.5 | Struktureller Aufbau einer Serie | 25 |
| 4.6 | Darstellung zwei Mitwirkender und eines Produktionsortes | 27 |
| 4.7 | Angabe eines Rechtes für eine Sendung | 27 |
| 4.8 | Darstellung eines Programmplanes | 28 |
| 4.9 | Abbildung der Essenz für ein Produktionselement | 29 |
| 4.10 | Vererbungshierarchie im Paket <i>Planning</i> | 31 |
| 4.11 | Ein Produktionselement <i>Scene</i> , das inhaltlich genehmigt wurde | 32 |
| 4.12 | Darstellung eines Sendeplans | 33 |
| 4.13 | Angabe einer Bildbeschreibung für eine Sendung | 34 |
| 4.14 | Vererbungshierarchie von <i>DMData</i> | 35 |

| | | |
|------|---|----|
| 4.15 | Angabe von Zusatzinformationen zu einer Sendung | 35 |
| 4.16 | Darstellung eines auf einem Festival ausgezeichneten Mitwirkenden | 36 |
| 4.17 | Zuweisung einer Kategorie für eine Sendung | 37 |
| 4.18 | Verschlagwortung einer Sendung | 37 |
| | | |
| 6.1 | Beispiel zur Klassennotation des Mappings | 44 |
| 6.2 | Sonderfall zur Klassennotation des Mappings | 44 |
| 6.3 | Die Klassen <i>SinglePEFull</i> und <i>SinglePEStandard</i> | 46 |
| 6.4 | Die Klassen <i>PEFull</i> und <i>PEStandard</i> | 47 |
| 6.5 | Hierarchische Abbildung einer <i>DU</i> -Instanz auf eine BMF-Instanz | 48 |
| 6.6 | Abbildung eines <i>SendeHaupttitels</i> | 56 |
| 6.7 | Abbildung von zwei Teilnehmern einer Produktion | 57 |
| 6.8 | Abbildung von <i>CONTENT</i> im Falle von Sachinhalt | 59 |
| 6.9 | Abbildung von <i>CONTENT</i> | 60 |
| 6.10 | Abbildung der Tabelle <i>CATEGORY</i> | 62 |
| 6.11 | Abbildung der Tabelle <i>MD</i> | 64 |
| 6.12 | Abbildung der Tabelle <i>REMARK</i> | 66 |
| 6.13 | Abbildung der Tabelle <i>INFO</i> | 67 |
| 6.14 | Abbildung der Tabelle <i>LINK</i> | 68 |
| 6.15 | Abbildung von <i>LICENCE</i> auf <i>Rights</i> | 70 |
| 6.16 | Abbildung von <i>ID-OF-DU</i> | 71 |
| 6.17 | Abbildung von Ausstrahlungsdaten | 75 |
| 6.18 | Abbildung von Rechtesequenzen | 78 |
| 6.19 | Referenzierung auf ein Lizenzobjekt | 81 |
| 6.20 | Die Klassen <i>FullOrder</i> und <i>StandardOrder</i> | 82 |
| 6.21 | Abbildung einer <i>EDL</i> | 85 |

Tabellenverzeichnis

| | | |
|------|--|----|
| 4.1 | Die Paketstruktur von BMF | 23 |
| 6.1 | Hierarchische Abbildungen | 48 |
| 6.2 | Abbildung der Tabelle <i>DU</i> | 50 |
| 6.3 | Abbildung des BR-Automationsdatensatzes | 53 |
| 6.4 | Attribut <i>Materialkategorie</i> | 54 |
| 6.5 | <i>TYPE</i> -Werte der Tabelle <i>TITLE</i> und deren Auswirkung auf das Mapping | 55 |
| 6.6 | Mapping der Tabelle <i>TITLE</i> | 55 |
| 6.7 | Abbildung der Variable <i>TYPE</i> | 58 |
| 6.8 | Abbildung der Tabelle <i>APC</i> | 58 |
| 6.9 | Abbildung der Tabelle <i>CONTENT</i> im Falle Sachinhalt | 60 |
| 6.10 | Abbildung der Tabelle <i>CONTENT</i> | 61 |
| 6.11 | Abbildung der Tabelle <i>CATEGORY</i> | 63 |
| 6.12 | Abbildung der Tabelle <i>MD</i> | 64 |
| 6.13 | Attribut <i>TYPE</i> der Tabelle <i>REMARK</i> | 65 |
| 6.14 | Abbildung der Tabelle <i>INFO</i> | 67 |
| 6.15 | Abbildung der Tabelle <i>LINK</i> | 69 |
| 6.16 | Abbildung <i>LICENCE</i> auf <i>Rights</i> | 70 |
| 6.17 | Abbildung der Tabelle <i>ID-OF-DU</i> | 72 |
| 6.18 | Bedeutungsdimensionen des Attributes <i>TYPE</i> in <i>BC</i> | 72 |
| 6.19 | Abbildung <i>BC</i> auf <i>Transmission</i> | 74 |
| 6.20 | Abbildung der Tabelle <i>RESTRICTION</i> | 78 |
| 6.21 | Abbildung zeitliche Parameter Rechtsequenzen | 79 |
| 6.22 | Abbildung der BR-Tabelle <i>Order</i> (Kopfdaten) | 83 |
| 6.23 | Abbildung einer EDL | 84 |
| 6.24 | Abbildung eines Signals einer EDL | 86 |

| | | |
|------|---|----|
| 6.25 | Abbildung eines Schnittes einer EDL | 87 |
|------|---|----|